(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 881 598 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.12.1998 Bulletin 1998/49

(51) Int. Cl.^{6}: G06T 11/40

(21) Application number: 97117786.0

(22) Date of filing: 14.10.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV RO SI

(30) Priority: 27.05.1997 JP 136693/97

(71) Applicant:
MITSUBISHI DENKI KABUSHIKI KAISHA
Chiyoda-ku Tokyo (JP)

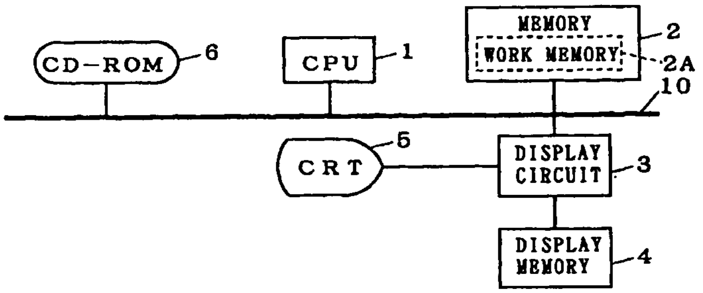
(72) Inventor: Takita, Hiroki
Chiyoda-ku, Tokyo 100 (JP)

(74) Representative:
Pellmann, Hans-Bernd, Dipl.-Ing. et al
Patentanwaltsbüro
Tiedtke-Bühling-Kinne & Partner
Bavariaring 4
80336 München (DE)

(54) Method for filling a polygon and image processing system

(57) For a polygon without self-intersection, with one of line segments constituting the outline of the polygon selected sequentially in the counterclockwise rotation, all the bits in the right direction of the selected line segment on the horizontal coordinate, inclusive of those for the selected line segment, on the plane work area (1 bit/pixel) of the work memory are inverted to draw (Step S22). Subsequently, only some of the line segments constituting the outline of the polygon whose vectors are in the upward direction (which makes an upward vertical-coordinate variation) are further drawn on the work memory (2A) (Step S23). Finally, the drawing information is transferred with extension from the work memory to the display memory (Step S24). Having the above steps, a method for achieving proper filling of a polygon with a decrease in load on a processor is provided.

FIG. 1



Printed by Xerox (UK) Business Services
2.16.6/3.4

Description

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a method for filling a polygon to be displayed in computer graphics based on the information stored in a display memory.

Description of the Background Art

Filling of a polygon in computer graphics has been implemented by using a scan line method and the like in the background art. This is a method in which a polygon is scanned from top to bottom with a horizontal line called a scan line to draw according to the following steps:

1. Take the scan line at the top of a polygon,
2. Repeat the following steps until the scan line reaches the bottom;

   2-1. Obtain sides which intersect the scan line,
   2-2. Obtain points of intersection of the scan line and these sides,
   2-3. Sort the points of intersection according to x-coordinate,
   2-4. Make pairs of two points of intersection in ascending order of x-coordinate and fill the two through,
   2-5. Move to a lower scan line by one,

3. End.

In the step 2-2 of obtaining points of intersection of the scan line and these sides, when a point of intersection is a vertex of a polygon and the vectors of two sides have upward and downward directions, an exceptional process is performed, assuming that there are two points of intersection at the vertex. With respect to an object of Fig. 59, the exceptional process is performed on the points at the coordinates (S1, H4), (S2, H11), (S5, H8), (S6, H7), (S8, H5) and (S10, H11).

For example, when the line segments in the object of Fig. 59 are selected in a counterclockwise rotation as shown in Fig. 60, as the line-segment vector A1 (H4-H8, S1-S5) is in an upward direction (which makes an upward vertical-coordinate variation) and the line-segment vector A2 (H1-H4, S4-S1) is in a downward direction (which makes a downward vertical-coordinate variation), it is assumed that there are two points of intersection at the coordinate (H4, S1) having two sides of the vectors A1 and A2.

On the other hand, another method for filling a polygon is an outline font method, using a drawing work memory of 1 bit/pixel. This is a method in which the bits for the line segments of a polygon and in their right directions are inverted in a work memory (1 bit/pixel) having a minimum rectangular which surrounds an area to be drawn to make a drawing in the area of 1 bit/pixel and the drawing information in the work memory is transferred with extension to a display memory to draw a filled polygon thereon.

In the outline font method, based on outline information defining line segments constituting the outline of a polygon to be filled, selecting a line segment in a predetermined rotation, all the bits for the selected line segment and in its right direction on horizontal coordinate on the work memory are inverted to draw a filled object on the work memory. This process is performed according to the following conditions A) to G):

A) When both a line segment to be now subjected to the bit inversion and a line segment to be next have upward vectors (which makes an upward vertical-coordinate variation), the bits in the right direction of the line segment inclusive of the initial point and not inclusive of the terminal point are inverted.
B) When both a line segment to be now subjected to the bit inversion and a line segment to be next have downward vectors (which makes a downward vertical-coordinate variation), the bits in the right direction of the line segment inclusive of the initial point and not inclusive of the terminal point are inverted.
C) When a line segment to be now subjected to the bit inversion has an upward vector and a line segment to be next has a downward vector, the bits in the right direction of the line segment inclusive of the initial point and the terminal point are inverted.
D) When a line segment to be now subjected to the bit inversion has a downward vector and a line segment to be next has an upward vector, the bits in the right direction of the line segment inclusive of the initial point and the terminal point are inverted.
E) When a line segment to be now subjected to the bit inversion has a downward vector and a line segment to be next has a horizontal vector (which makes no vertical-coordinate variation), the bits in the right direction of the line segment inclusive of the initial point and not inclusive of the terminal point are inverted.
F) When a line segment to be now subjected to the bit inversion has an upward vector and a line segment to be next has a horizontal vector (which makes no vertical-coordinate variation), the bits in the right direction of the line segment inclusive of the initial point and not inclusive of the terminal point are inverted.
G) When a line segment to be now subjected to the bit inversion has a horizontal vector and a line segment to be next has a downward vector, the bits in the right direction of the line segment inclusive of the initial point and not inclusive of the terminal point are inverted.

H) When a line segment to be now subjected to the bit inversion has a horizontal vector and a line segment to be next has an upward vector, the bits in the right direction of the line segment inclusive of the initial point and not inclusive of the terminal point are inverted.

The outline font method will be discussed below, taking a triangle to be filled having the outline shown in Fig. 61 as an example. First, the coordinates of three vertices {P1(H8, S2), P2(H2, S15), P3(H16, S11)} are obtained as the outline information.

With respect to a triangle to be filled, the line-segment vectors L1 (H2-H8, S15-S2), L2 (H16-H2, S11-S15) and L3 (H8-H16, S2-S11) are selected in this order in a counterclockwise rotation as shown in Fig. 62. A plane work area on the work memory is blank (nothing is drawn) in an initial state.

First, the line segment L1 is obtained from the coordinates of the vertices P1 and P2 given as the outline information, using an already-existing linear algorithm such as Bresenham algorithm, as shown in Figs. 63 and 64, and all the bits in the right direction of the line segment L1 on the horizontal coordinate on the plane work area (1 bit/pixel) of the work memory are inverted for drawing.

Fig. 63 shows a halfway state of the bit inversion of the line segment L1 and Fig. 64 shows the final result of the bit inversion. As the line segment L1 is in the downward direction and the line segment L2 is in the upward direction, according to the condition D), the bits in the right direction of the line segment L1 inclusive of the initial point and the terminal point are inverted.

Secondly, the line segment L2 is obtained from the coordinates of the vertices P2 and P3 given as the outline information, using the already-existing linear algorithm such as Bresenham algorithm, as shown in Fig. 65 and all the bits in the right direction of the line segment L2 on the horizontal coordinate on the plane work area (1 bit/pixel) of the work memory are inverted. As a result, some of the already-drawn bits of Fig. 64 are inverted again to be back in blank.

Fig. 65 shows a halfway state of the bit inversion of the line segment L2. As the line segments L2 and L3 are both in the upward direction, according to the condition A), the bits in the right direction of the line segment L2 inclusive of the initial point and not inclusive of the terminal point are inverted.

Similarly, the line segment L3 is obtained from the outline information, and all the bits in the right direction of the line segment L3 on the horizontal coordinate on the plane work area (1 bit/pixel) of the work memory are inverted, to complete the triangle which is filled inside as shown in Fig. 66. As the line segment L3 is in the upward direction and the line segment L1 is in the downward direction, according to the condition C), the bits in the right direction of the line segment L3 inclusive of the initial point and the terminal point are inverted.

After that, the drawing information on the plane work area of the work memory is transferred with extension to a plane display area of the display memory and the triangle which is filled inside is displayed, based on the information stored in the display memory. "Extension" refers to addition of another information such as color information.

Furthermore, in the above outline font method, though the bits in the right direction of each line segment of a polygon are inverted, it is possible, on the same principle, that the bits in the left direction of each line segment of the polygon are inverted.

When it is intended that a large object is drawn through this method, the object is divided into parts so that each of the parts can be processed on the plane work area of the work memory. The divided parts are filled on the respective plane work areas of the work memory and then sequentially transferred to the display memory. That allows filling of a large object even with a work memory having small plane work areas.

The background-art computer graphics adopts the scan line method or the outline font method as discussed above.

The scan line method needs a sort of points of intersection at every raster (scan line) and a relatively large memory to store information on line segments (initial point, terminal point, slope and the like).

As the number of vertices increases, the number of memories increases since memories as much as vertices are needed and also the load on the processor increases with an increase in the number of points of intersection to be sorted.

On the other hand, the outline font method is not likely to have the above disadvantage but has another problem of inaccuracy of filling due to 1-bit (1-pixel) lack on the right side (lack on the left side in a case of the bit inversion in the left direction) of the drawn polygon in drawing a polygon on the 1-bit/pixel plane work area of the work memory according to the conditions A) to H).

SUMMARY OF THE INVENTION

The present invention is directed to a method for filling a polygon, using an image processing system which has a work memory having a rectangular memory area of a plurality of bits, for storing information of one pixel by n (n ≧ 1) bits and a display memory having a rectangular memory area of a plurality of bits, for storing information of one pixel by m (m > n) bits and displays an image based on the information stored in the display memory, to fill the inside of the polygon. The work memory is blank in an initial state. According to a first aspect of the present invention, the method comprises the steps of: (a) obtaining outline information defining line segments constituting an outline of a polygon to be filled; (b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of the work memory corresponding to each of the line segments while selecting each of the line segments constituting the outline of the polygon to be filled in a predetermined rotation based on the outline information; (c) further drawing the line segments constituting the outline of the polygon to be filled on the work memory where the drawing of filled polygon is performed based on the outline information; and (d) transferring information stored in the work memory to the display memory after the step (c).

According to a second aspect of the present invention, in the method of the first aspect, the step (c) includes the step of drawing all the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation.

According to a third aspect of the present invention, in the method of the first aspect, the polygon to be filled includes a polygon without self-intersection, the bit inversion in the step (b) includes one of a first inversion mode for inverting bits inclusive of those corresponding to the line segments on the work memory and a second inversion mode for inverting bits not inclusive of those corresponding to the line segments, the step (c) includes the step of drawing only some of the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in a predetermined vertical direction, and the predetermined vertical direction is determined by the bit inversion, the predetermined rotation and the predetermined horizontal direction, in which some of the line segments which fail to be drawn, lacking, in the bit inversion of the step (b) have a vertical variation.

According to a fourth aspect of the present invention, the method for filling a polygon comprises the steps of: (a) obtaining outline information defining line segments constituting an outline of a polygon without self-intersection to be filled; and (b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of the work memory corresponding to each of the line segments while selecting each of the line segments constituting the outline of the polygon to be filled in a predetermined rotation based on the outline information. In the method of the fourth aspect, the bit inversion includes one of a first inversion mode for inverting bits inclusive of those corresponding to the line segments on the work memory and a second inversion mode for inverting bits not inclusive of those corresponding to the line segments. According to the fourth aspect, the method further comprises the steps of; (c) transferring information stored in the work memory to the display memory after the step (b); and (d) drawing only some of the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in a predetermined vertical direction on the display memory based on the outline information after the step (c). In the method of the fourth aspect, the predetermined vertical direction is determined by the bit inversion, the predetermined rotation and the predetermined horizontal direction, in which some of the line segments which fail to be drawn, lacking, in the bit inversion of the step (b) have a vertical variation.

According to a fifth aspect of the present invention, in the method of the third or fourth aspect, the bit inversion includes the first inversion mode, the predetermined rotation includes a counterclockwise rotation, the predetermined horizontal direction includes a right direction, and the predetermined vertical direction includes an upward direction.

According to a sixth aspect of the present invention, in the method of the third or fourth aspect, the bit inversion includes the first inversion mode, the predetermined rotation includes a clockwise rotation, the predetermined horizontal direction includes a right direction, and the predetermined vertical direction includes a downward direction.

According to a seventh aspect of the present invention, in the method of the third or fourth aspect, the bit inversion includes the second inversion mode, the predetermined rotation includes a counterclockwise rotation, the predetermined horizontal direction includes a right direction, and the predetermined vertical direction includes a downward direction.

According to an eighth aspect of the present invention, in the method of the third or fourth aspect, the bit inversion includes the second inversion mode, the predetermined rotation includes a clockwise rotation, the predetermined horizontal direction includes a right direction, and the predetermined vertical direction includes an upward direction.

According to a ninth aspect of the present invention, the method for filling a polygon comprises the steps of: (a) obtaining outline information defining line segments constituting an outline of a polygon without self-intersection to be filled; (b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of the work memory corresponding to each of the line segments while selecting each of the line segments constituting the outline of the polygon to be filled in a predetermined rotation based on the outline information; (c) transferring information stored in the work memory to the display memory after the step (b). In the method of the ninth aspect, the bit inversion includes a first inversion mode for inverting bits inclusive of those corresponding to the line segments on the work memory and a second inversion mode for inverting bits not inclusive of those corresponding to the line segments, and some of the line segment constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in a first vertical direction are subjected to the bit inversion in the first inversion mode and some whose vectors have a vertical variation in a second vertical direction are subjected to the bit inversion in the second inversion mode, and the first and second vertical directions is determined by the predetermined rotation and the predetermined horizontal direction so that some of the line segments whose vectors have a vertical variation in the first vertical direction can be drawn with no lack when the bit inversion is performed in the first inversion mode and some of the line segments whose vectors have a vertical variation in the second vertical direction can be drawn with no lack when the bit inversion is performed in the second inversion mode.

According to a tenth aspect of the present invention, in the method of the ninth aspect, the predetermined rotation includes a counterclockwise rotation, the predetermined horizontal direction includes a right direction, the first predetermined vertical direction includes a downward direction, and the second predetermined vertical direction includes an upward direction.

According to an eleventh aspect of the present invention, in the method of the ninth aspect, the predetermined rotation includes a clockwise rotation, the predetermined horizontal direction includes a right direction, the first predetermined vertical direction includes an upward direction, and the second predetermined vertical direction includes a downward direction.

The present invention is also directed to a computer-readable recording medium for recording a program. According to a twelfth aspect of the present invention, the computer-readable recording medium records the method for filling a polygon, using an image processing system which has a work memory having a rectangular memory area of a plurality of bits, for storing information of one pixel by n (n ≧ 1) bits and a display memory having a rectangular memory area of a plurality of bits, for storing information of one pixel by m (m > n) bits and displays an image based on the information stored in the display memory, to fill the inside of the polygon, under a control of a computer. The work memory is blank in an initial state. The recording medium records a program to perform the steps of (a) obtaining outline information defining line segments constituting an outline of a polygon to be filled; (b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of the work memory with respect to each of the line segments while selecting each of the line segments constituting the outline of the polygon to be filled in a predetermined rotation based on the outline information; (c) further drawing the line segments constituting the outline of the polygon to be filled on the work memory where the drawing of filled polygon is performed based on the outline information; and (d) transferring information stored in the work memory to the display memory after the step (c).

According to a thirteenth aspect of the present invention, the computer-readable recording medium records the method for filling a polygon, using an image processing system which has a work memory having a rectangular memory area of a plurality of bits, for storing information of one pixel by n (n ≧ 1) bits and a display memory having a rectangular memory area of a plurality of bits, for storing information of one pixel by m (m > n) bits and displays an image based on the information stored in the display memory, to fill the inside of the polygon, under a control of a computer. The work memory is blank in an initial state. The recording medium records a program to perform the steps of (a) obtaining outline information defining line segments constituting an outline of a polygon without self-intersection to be filled; and (b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of the work memory with respect to each of the line segments while selecting each of the line segments constituting the outline of the polygon to be filled in a predetermined rotation based on the outline information. In the method, the bit inversion includes one of a first inversion mode for inverting bits inclusive of those for the line segments on the work memory and a second inversion mode for inverting bits not inclusive of those for the line segments. According to the thirteenth aspect, the recording medium records the program to further perform the steps of; (c) transferring information stored in the work memory to the display memory after the step (b); and (d) drawing only some of the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in a predetermined vertical direction on the display memory based on the outline information after the step (c). In the method, the predetermined vertical direction depends on the bit inversion, the predetermined rotation and the predetermined horizontal direction, in which some of the line segments which fail to be drawn, lacking, in the bit inversion of the step (b) have a vertical variation.

According to a fourteenth aspect of the present invention, the computer-readable recording medium records the method for filling a polygon, using an image processing system which has a work memory having a rectangular memory area of a plurality of bits, for storing information of one pixel by n (n ≧ 1) bits and a display memory having a rectangular memory area of a plurality of bits, for storing information of one pixel by m (m > n) bits and displays an image based on the information stored in the display memory, to fill the inside of the polygon, under a control of a computer. The work memory is blank in an initial state. The recording medium records a program to perform the steps of (a) obtaining outline information defining line segments constituting an outline of a polygon without self-intersection to be filled; (b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of the work memory with respect to each of the line segments while selecting each of the line segments constituting the outline of the polygon to be filled in a predetermined rotation based on the outline information; (c) transferring information stored in the work memory to the display memory after the step (b). In the method, the bit inversion includes a first inversion mode for inverting bits inclusive of those for the line segments on the work memory and a second inversion mode for inverting bits not inclusive of those for the line segments, and some of the line segment constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in a first vertical direction are subjected to the bit inversion in the first inversion mode and some whose vectors have a vertical variation in a second vertical direction are subjected to the bit inversion in the second inversion mode, and the first and second vertical directions depend on the predetermined rotation and the predetermined horizontal direction so that some of the line segments whose vectors have a vertical variation in the first vertical direction can be drawn with no lack when subjected to the bit inversion in the first inversion mode and some of the line segments whose vectors have a vertical variation in the second vertical direction can be drawn with no lack when subjected to the bit inversion in the second inversion mode.

In accordance with the method for filling a polygon of the first aspect of the present invention, in the step (c), the line segments constituting the outline of the polygon to be filled are further drawn based on the outline information on the work memory where the drawing of filled polygon is performed, and therefore a properly-filled polygon can be obtained even if an incompletely-filled polygon with some of the line segments constituting the outline lacking is drawn in the step (b).

Moreover, as the work memory has smaller number of bits for one pixel than the display memory, performing the step (c) on the work memory reduces the number of memory accesses and suppresses the load on the CPU, thus achieving faster drawing of filled polygon.

In accordance with the method of the second aspect of the present invention, since all the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation are drawn in the step (c), a properly-filled polygon can be surely obtained from an incompletely-filled polygon with some of the line segments constituting the outline lacking.

In accordance with the method of the third aspect of the present invention, in the step (c), only some of the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in the second vertical direction. The second direction is a direction in which some of the line segments which fail to be drawn, lacking, in the bit inversion of the step (b) have a vertical variation.

Therefore, instead of all the line segments constituting the outline of the polygon, only some of the line segments whose vectors are in the second vertical direction, which are needed for proper filling of the polygon, are efficiently drawn in the step (c), and much faster drawing of filled polygon is thereby realized.

In accordance with the method of the fourth aspect of the present invention, in the step (d), only some of the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in the second vertical direction. The second direction is a direction in which some of the line segments which fail to be drawn, lacking, in the bit inversion of the step (b) have a vertical variation

Therefore, instead of all the line segments constituting the outline of the polygon, only some of the line segments whose vectors are in the second vertical direction, which are needed for proper filling of the polygon, are efficiently drawn in the step (d), and much faster drawing of filled polygon is thereby realized.

In accordance with the method of the fifth aspect of the present invention, when the bit inversion is performed in the first inversion mode to invert bits inclusive of those corresponding to the line segments on the work memory, the predetermined rotation is the counterclockwise rotation and the predetermined horizontal direction is the right direction, the second vertical direction is uniquely determined to be the upward direction.

In accordance with the method of the sixth aspect of the present invention, when the bit inversion is performed in the first inversion mode to invert bits inclusive of those corresponding to the line segments on the work memory, the predetermined rotation is the clockwise rotation and the predetermined horizontal direction is the right direction, the second vertical direction is uniquely determined to be the downward direction.

In accordance with the method of the seventh aspect of the present invention, when the bit inversion is performed in the second inversion mode to invert bits not inclusive of those corresponding to the line segments on the work memory, the predetermined rotation is the counterclockwise rotation and the predetermined horizontal direction is the right direction, the second vertical direction is uniquely determined to be the downward direction.

In accordance with the method of the eighth aspect of the present invention, when the bit inversion is performed in the second inversion mode to invert bits not inclusive of those corresponding to the line segments on the work memory, the predetermined rotation is the clockwise rotation and the predetermined horizontal direction is the right direction, the second vertical direction is uniquely determined to be the upward direction.

In accordance with the method of the ninth aspect of the present invention, the bit inversion of the step (b) includes the first inversion mode for inverting bits inclusive of those corresponding to the line segments on the work memory and the second inversion mode for inverting bits not inclusive of those corresponding to the line segments, and some of the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in the first vertical direction are subjected to the bit inversion in the first inversion mode and some whose vectors have a vertical variation in the second vertical direction are subjected to the bit inversion in the second inversion mode.

Further, the first and second vertical directions are determined so that some of the line segments whose vectors have a vertical variation in the first vertical direction can be drawn with no lack when the bit inversion is performed in the first inversion mode and some of the line segments whose vectors have a vertical variation in the second vertical direction can be drawn with no lack when the bit inversion is performed in the second inversion mode.

As a result, since a proper filling of the polygon can be achieved by only one operation of the step (b), much faster drawing of filled polygon is realized.

Moreover, as the work memory has smaller number of bits for one pixel than the display memory, performing the step (b) on the work memory reduces the number of memory accesses and suppresses the load on the CPU, thus achieving faster drawing of filled polygon.

In accordance with the method of the tenth aspect of the present invention, when the predetermined rotation is the counterclockwise rotation and the predetermined horizontal direction is the right direction, the first and second vertical directions are uniquely determined to be the downward and upward directions, respectively.

In accordance with the method of the eleventh aspect of the present invention, when the predetermined rotation is the clockwise rotation and the predetermined horizontal direction is the right direction, the first and second vertical directions are uniquely determined to be the upward and downward directions, respectively.

In the step (c) performed in accordance with the program recorded in the recording medium of the twelfth aspect of the present invention, the line segments constituting the outline of the polygon to be filled are further drawn based on the outline information on the work memory where the drawing of filled polygon is performed, and a properly-filled polygon can be therefore obtained even if an incompletely-filled polygon with some of the line segments constituting the outline lacking is drawn in the step (b).

In the step (d) performed in accordance with the program recorded in the recording medium of the thirteenth aspect of the present invention, only some of the line segments constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in the second vertical direction are drawn. The second direction is a direction in which some of the line segments which fail to be drawn, lacking, in the bit inversion of the step (b) have a vertical variation.

Therefore, instead of all the line segments constituting the outline of the polygon, only some of the line segments whose vectors are in the second vertical direction, which are needed for proper filling of the polygon, are efficiently drawn in the step (c), and much faster drawing of filled polygon is thereby realized.

In the step (b) performed in accordance with the program recorded in the recording medium of the fourteenth aspect of the present invention, the bit inversion includes the first inversion mode for inverting bits inclusive of those corresponding to the line segments on the work memory and the second inversion mode for inverting bits not inclusive of those corresponding to the line segments, and some of the line segment constituting the outline of the polygon to be filled and selected in the predetermined rotation whose vectors have a vertical variation in the first vertical direction are subjected to the bit inversion in the first inversion mode and some whose vectors have a vertical variation in the second vertical direction are subjected to the bit inversion in the second inversion mode.

Further, the first and second vertical directions are determined so that some of the line segments whose vectors have a vertical variation in the first vertical direction can be drawn with no lack when subjected to the bit inversion in the first inversion mode and some of the line segments whose vectors have a vertical variation in the second vertical direction can be drawn with no lack when subjected to the bit inversion in the second inversion mode.

As a result, since a proper filling of the polygon can be achieved by only one operation of the step (b) performed in accordance with the program recorded in the recording medium of the fourteenth aspect, much faster drawing of filled polygon is realized.

Moreover, as the work memory has smaller number of bits for one pixel than the display memory, performing the step (b) on the work memory reduces the number of memory accesses and suppresses the load on the CPU, thus achieving faster drawing of filled polygon.

An object of the present invention is to provide a method for accurate filling of a polygon with a decrease in the load on a processor.

These and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing a hardware construction used in all preferred embodiments of the present invention;

Figs. 2 to 4 are block diagram showing another hardware constructions;

Fig. 5 is a flow chart of process for filling a polygon in accordance with a method of a first preferred embodiment;

Fig. 6 is a flow chart of process for filling a polygon in accordance with a method of a second preferred embodiment;

Fig. 7 is a flow chart of process for filling a polygon in accordance with a method of a third preferred embodiment;

Fig. 8 is a flow chart of process for filling a polygon in accordance with a method of a fourth preferred embodiment;

Fig. 9 is a flow chart of process for filling a polygon in accordance with a method of a fifth preferred embodiment;

Fig. 10 is a flow chart of process for filling a polygon in accordance with a method of a sixth preferred embodiment;

Fig. 11 is a flow chart of process for filling a polygon in accordance with a method of a seventh preferred embodiment;

Fig. 12 is an illustration of a polygon with self-intersection;

Fig. 13 is an illustration of the method for filling a polygon in accordance with the first preferred embodiment of the present invention;

Fig. 14 is a schematic view of the method for filling a polygon in accordance with the first preferred embodiment;

Figs. 15 to 19 are illustrations of the method for filling a polygon in accordance with the first preferred embodiment;

Figs. 20 to 25 are illustrations of the method for filling a polygon in accordance with the second preferred embodiment of the present invention;

Figs. 26 to 31 are illustrations of the method for filling a polygon in accordance with the third preferred embodiment of the present invention;

Fig. 32 is an illustration of the method for filling a polygon in accordance with the fourth preferred embodiment of the present invention;

Fig. 33 is a schematic view of the method for filling a polygon in accordance with the fourth preferred embodiment;

Figs. 34 to 38 are illustrations of the method for filling a polygon in accordance with the fourth preferred embodiment of the present invention;

Figs. 39 to 44 are illustrations of the method for filling a polygon in accordance with the fifth preferred embodiment of the present invention;

Figs. 45 to 47 are illustrations of the method for filling a polygon in accordance with the sixth preferred embodiment of the present invention;

Figs. 48 and 49 are illustrations of areas to be inverted according to a second inversion mode;

Figs. 50 to 51 are illustrations of the method for filling a polygon in accordance with the sixth preferred embodiment of the present invention;

Figs. 52 and 53 are illustrations of the method for filling a polygon in accordance with the seventh preferred embodiment of the present invention;

Figs. 54 and 55 are illustrations of areas to be inverted according to a second inversion mode;

Figs. 56 to 58 are illustrations of the method for filling a polygon in accordance with the seventh preferred embodiment of the present invention;

Fig. 59 is an illustration of a background-art scan line method;

Fig. 60 is a schematic view of the background-art scan line method;

Fig. 61 is an illustration of a background-art outline font method;

Fig. 62 is a schematic view of the background-art outline font method;

and

Figs. 63 to 66 are illustrations of the background-art outline font method.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

⟨Hardware Construction⟩

Fig. 1 is a block diagram showing a hardware construction of an image processing system used for methods for filling a polygon in accordance with the first to seventh preferred embodiments. As shown in this figure, the image processing system comprises a CPU 1, a memory 2, a display circuit 3, a display memory 4, a CRT 5 and a CD-ROM 6, and the CPU 1, the memory 2, the display circuit 3 and the CD-ROM 6 perform data transfer through a bus 10.

The CPU 1 reads a program recording a process in accordance with the method for filling a polygon as discussed later from the CD-ROM 6. According to the program, the CPU 1 performs filling of a polygon, using a rectangular plane work area of a work memory 2A allocated in the memory 2 and eventually transfers drawing information stored in the work memory 2A of the memory 2 to a rectangular plane display area in the display memory 4 through the bus 10 and the display circuit 3 with extension. The display circuit 3 displays an image on the CRT 5 based on the drawing information stored in the display memory 4.

While the work memory 2A stores information of 1 bit/pixel, the display memory 4 generally stores information of multibit per pixel such as 4 bits/pixel (for 16-color display) and 8 bits/pixel (for 256-color display). Accordingly, it is needed that data of 1 bit/pixel in the work memory 2A should be extended to data of 4 bits/pixel or 8 bits/pixel, with color information added when transferred to the display memory 4.

Then, the CPU 1 transfers the drawing information on the work memory 2A, being extended with another information such as color information added, to the display memory 4. Furthermore, the color information is stored in a predetermined area in the memory 2 or in the display circuit 3.

With construction as shown in Fig. 2, data transfer between the memory 2 and the display memory 4 may be performed through only the bus 10.

As shown in Figs. 3 and 4, a work memory 7A and a display memory 7B may be incorporated in a common memory 7. Fig. 3 shows a construction in which the common memory 7 is connected direct to the display circuit 3 and Fig. 4 shows a construction in which the common memory 7 is connected to the bus 10. Further, in the construction of Fig. 3, data transfer with extension from the work memory 7A to the display memory 7B in the common memory 7 may be performed under control of the display circuit 3.

⟨Principles of The Preferred Embodiments⟩

In the outline font method using a 1-bit/pixel drawing work memory, the image processing system having one of the constructions of Figs. 1 to 4 performs the following steps in sequential, to fill a polygon.

[Principle of The First Preferred Embodiment]

As shown in Fig. 5, the following steps are performed: First, the outline information (coordinates of vertices) defining line segments constituting the outline of a polygon to be filled is obtained (Step S11). Then, the bit inversion is performed on the work memory of 1-bit/pixel according to the conditions A) to H), discussed in the background art, to fill the polygon (Step S12). Further, the frame of the polygon (all the line segments constituting the outline) is drawn (Step S13). Finally, the drawing information is transferred with extension from the work memory to the display memory (Step S14).

[Principle of The Second Preferred Embodiment]

As shown in Fig. 6, the following steps are performed: First, the outline information is obtained (Step S21). Then, for a polygon without self-intersection, with one of the line segments selected sequentially in the counterclockwise rotation (in a predetermined rotation), the bit inversion is performed on the work memory of 1-bit/pixel according to the conditions A) to H) to fill the polygon (Step S22). Subsequently, some of the line segments whose vectors are in the upward direction (which makes an upward vertical-coordinate variation; in a predetermined vertical direction) are drawn on the work memory (Step S23). Finally, the drawing information is transferred with extension from the work memory to the display memory (Step S24).

The polygon without self-intersection refers to a polygon having no point of intersection but the vertices, and does not include a shape such as the star shape of Fig. 12 constituted of vertices P11 to P15 that makes self-intersection with points of intersection C1 to C5.

[Principle of The Third Preferred Embodiment]

As shown in Fig. 7, the following steps are performed: First, the outline information is obtained (Step S31). Then, for a polygon without self-intersection, with one of the line segments selected sequentially in the counterclockwise rotation (in a predetermined rotation), the bit inversion is performed on the work memory of 1-bit/pixel according to the conditions A) to H) to fill the polygon (Step S32). Subsequently, the drawing information is transferred with extension from the work memory to the display memory (Step S33). Finally, some of the line segments whose vectors are in the upward direction (in a predetermined vertical direction) are drawn on the display memory (Step S34).

[Principle of The Fourth Preferred Embodiment]

As shown in Fig. 8, the following steps are performed: First, the outline information is obtained (Step S41). Then, for a polygon without self-intersection, with one of the line segments selected sequentially in the clockwise rotation (in a predetermined rotation), the bit inversion is performed on the work memory of 1-bit/pixel according to the conditions A) to H) to fill the polygon (Step S42). Subsequently, some of the line segments whose vectors are in the downward direction (which makes a downward vertical-coordinate variation; in a predetermined vertical direction) are drawn on the work memory (Step S43). Finally, the drawing information is transferred with extension from the work memory to the display memory (Step S44).

[Principle of The Fifth Preferred Embodiment]

As shown in Fig. 9, the following steps are performed: First, the outline information is obtained (Step S51). Then, for a polygon without self-intersection, with one of the line segments selected sequentially in the clockwise rotation (in a predetermined rotation), the bit inversion is performed on the work memory of 1-bit/pixel according to the conditions A) to H) to fill the polygon (Step S52). Subsequently, the drawing information is transferred with extension from the work memory to the display memory (Step S53). Finally, some of the line segments whose vectors are in the downward direction (in a predetermined vertical direction) are drawn on the work memory (Step S54).

[Principle of The Sixth Preferred Embodiment]

As shown in Fig. 10, the following steps are performed: First, the outline information is obtained (Step S61). Then, for a polygon without self-intersection, with one of the line segments selected sequentially in the counterclockwise rotation (in a predetermined rotation), the bit inversion is performed on the work memory of 1-bit/pixel according to the conditions A) to H) to fill the polygon (Step S62). In this step, when the vector of a line segment is in the upward direction (in a second vertical direction), the bits not inclusive of those for the line segment are inverted, and when the vector of a line segment is in the other direction (in a first vertical direction), the bits inclusive of those for the line segment are inverted. After that, the drawing information is transferred with extension from the work memory to the display memory (Step S63).

[Principle of The Seventh Preferred Embodiment]

As shown in Fig. 11, the following steps are performed: First, the outline information is obtained (Step S71). Then, for a polygon without self-intersection, with one of the line segments selected sequentially in the clockwise rotation (in a predetermined rotation), the bit inversion is performed on the work memory of 1-bit/pixel according to the conditions A) to H) to fill the polygon (Step S72). In this step, when the vector of a line segment is in the downward direction (in a second vertical direction), the bits not inclusive of those for the line segment are inverted, and when the vector of a line segment is in the other direction (in a first vertical direction), the bits inclusive of those for the line segment are inverted. After that, the drawing information is transferred with extension from the work memory to the display memory (Step S73).

The First Preferred Embodiment

Figs. 13 to 19 are illustrations of the method for filling a polygon in accordance with the first preferred embodiment. Any of the hardware constructions of Figs. 1 to 4 may be used to implement the method for filling a polygon of the first preferred embodiment, and for explanatory convenience, the method using the hardware construction of Fig. 1 will be discussed below.

First, the CPU 1 reads a program recording a process in accordance with the method for filling a polygon of the first preferred embodiment from the CD-ROM 6 through the bus 10.

Then, the coordinates of three vertices {(P1 (H8, S2), P2 (H2, S15), P3 (H16, S11)} are externally obtained as the outline information of the polygon to be filled having the outline of Fig. 13 and are stored in a predetermined area of the memory 2 (Step S11 of Fig. 5). The plane work area of the work memory 2A in the memory 2 is blank (nothing is drawn) in an initial state. Hereinafter, in Fig. 13 illustrating the drawing, a blank square represents a not-drawn state and a solid square represents a drawn state.

In the method for filling a polygon of the first preferred embodiment, as shown in Fig. 14, for the triangle to be filled, the line segments L1 (H2-H8, S15-S2), L2 (H16-H2, S11-S15), and L3 (H8-H16, S2-S11) are selected in the order of the vertices P1, P2 and P3 in the counterclockwise rotation. The following operation is performed under the control of the CPU 1 based on the program read from the CD-ROM 6.

Next, the bit inversion is performed on the work memory 2A as shown in Figs. 15 to 17 (Step S12 of Fig. 5). The bit inversion will be discussed below.

First, as shown in Fig. 15, the line segment L1 is obtained from the coordinates of the vertices P1 and P2 given as the outline information, using an already-existing linear algorithm such as Bresenham algorithm, and all the bits in the right direction (in a predetermined horizontal direction) of the line segment L1 on the horizontal coordinate, inclusive of those corresponding to the line segment L1, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw. As the line segment (vector) L1 is in the downward direction (which makes a downward vertical-coordinate variation) and the line segment (vector) L2 is in the upward direction (which makes an upward vertical-coordinate variation), according to the condition D), the bits in the right direction of the line segment L1 inclusive of the initial point and the terminal point are inverted.

Secondly, as shown in Fig. 16, the line segment L2 is obtained from the coordinates of the vertices P2 and P3 given as the outline information, using the already-existing linear algorithm such as Bresenham algorithm, and all the bits in the right direction of the line segment L2 on the horizontal coordinate, inclusive of those corresponding to the line segment L2, on the plane work area of the work memory 2A are inverted. As a result, some of the already-drawn bits of Fig. 15 are inverted again to be back in blank.

Fig. 16 shows a halfway state of the bit inversion of the line segment L2. As the line segments L2 and L3 are both in the upward direction, according to the condition A), the bits in the right direction of the line segment L2 inclusive of the initial point and not inclusive of the terminal point are inverted.

Similarly, the line segment L3 is obtained based on the outline information, and all the bits in the right direction of the line segment L3 on the horizontal coordinate, inclusive of those corresponding to the line segment L3, on the plane work area of the work memory 2A are inverted. As the line segment L3 is in the upward direction and the line segment L1 is in the downward direction, according to the condition C), the bits in the right direction of the line segment L3 inclusive of the initial point and the terminal point are inverted. As a result, when the bit inversion is completed, the triangle which is filled inside as shown in Fig. 17 is achieved. But, it is an incompletely-filled triangle with some of the line segments on the right side lacking by bit (pixel).

Subsequently, as shown in Fig. 18, the frame of the triangle (represented by hatched squares in Fig. 18) constituted of the line segments L1 to L3 obtained based on the outline information is further drawn on the incompletely-filled triangle on the work memory 2A, to achieve a properly-filled triangle (Step S13 of Fig. 5).

After that, as shown in Fig. 19, the drawing information on the plane work area of the work memory 2A is transferred with extension to the plane display area of the display memory 4 through the bus 10 and the display circuit 3 (Step S14 of Fig. 5).

As a result, the triangle which is properly filled inside is displayed on the CRT 5 by the display circuit 3 based on the information stored in the display memory 4.

Thus, in the method for filling a polygon of the first preferred embodiment, the frame of the polygon is drawn on the work memory where the incompletely-filled polygon is drawn to properly fill the polygon, and then the drawing information is transferred with extension from the work memory to the display memory.

As mentioned earlier, for color display, the number of bits per pixel in the display memory is 4 bits/pixel, 8 bits/pixel or the like, more than that (1 bit/pixel) in the work memory.

As the amount of memories needed for one pixel is smaller in the work memory than in the display memory, drawing the frame of a polygon on the work memory as revealed in the first preferred embodiment reduces the number of memory accesses and suppresses the load on the CPU 1, thus achieving faster drawing of filled polygon.

Though the line segments are selected in the counterclockwise rotation in the above method for filling a polygon of the first preferred embodiment, the line segments may be selected in the clockwise rotation, to produce the same effect. In the first preferred embodiment, since all the line segments constituting the outline of a polygon to be filled and selected in the counterclockwise rotation or the clockwise rotation are drawn, the properly-filled polygon can be surely achieved from the incompletely-filled polygon with some of the line segments constituting the outline lacking which is obtained after the bit inversion.

Further, though the bits inclusive of those for the line segments are inverted in the above method for filling a polygon of the first preferred embodiment, the bits may be inverted non inclusive of those for the line segments.

Furthermore, though the bits in the right direction of the line segments constituting the polygon are inverted in the above method for filling a polygon of the first preferred embodiment, the bits in the left direction of the line segments constituting the polygon may be inverted, based on the same principle. In this case, by drawing the frame of the polygon, the properly-filled polygon can be surely achieved from the incompletely-filled polygon with some of the line segments on the left side lacking by bit (pixel).

The Second Preferred Embodiment

Figs. 20 to 25 are illustrations of the method for filling a polygon in accordance with the second preferred embodiment. Any of the hardware constructions of Figs. 1 to 4 may be used to implement the method for filling a polygon of the second preferred embodiment, and for explanatory convenience, the method using the hardware construction of Fig. 1 will be discussed below.

First, the CPU 1 reads a program recording a method for filling a polygon of the second preferred embodiment from the CD-ROM 6 through the bus 10. The method for filling a polygon of the second preferred embodiment is used for processing a polygon without self-intersection.

Then, the coordinates of three vertices {(P1 (H8, S2), P2 (H2, S15), P3 (H16, S11)} are externally obtained as the outline information of the polygon to be filled having the outline of Fig. 20 and are stored in a predetermined area of the memory 2 (Step S21 of Fig. 6). The plane work area of the work memory 2A in the memory 2 is blank in an initial state.

In the method for filling a polygon of the second preferred embodiment, as shown in Fig. 14, for the triangle to be filled, the line segments L1, L2 and L3 are selected in the order of the vertices P1, P2 and P3 in the counterclockwise rotation like in the first preferred embodiment. The following operation is performed under the control of the CPU 1 based on the program read from the CD-ROM 6.

Next, the bit inversion is performed on the work memory 2A as shown in Figs. 21 to 23 (Step S22 of Fig. 6). The bit inversion will be discussed below.

First, as shown in Fig. 21, the line segment L1 is obtained from the coordinates of the vertices P1 and P2 given as the outline information, using an already-existing linear algorithm such as Bresenham algorithm, and all the bits in the right direction (in a predetermined horizontal direction) of the line segment L1 on the horizontal coordinate, inclusive of those corresponding to the line segment L1, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw.

Secondly, as shown in Fig. 22, the line segment L2 is obtained from the coordinates of the vertices P2 and P3 given as the outline information, and all the bits in the right direction of the line segment L2 on the horizontal coordinate, inclusive of those corresponding to the line segment L2, on the plane work area of the work memory 2A are inverted. As a result, some of the already-drawn bits of Fig. 21 are inverted again to be back in blank.

Similarly, the line segment L3 is obtained based on the outline information, and all the bits in the right direction of the line segment L3 on the horizontal coordinate, inclusive of those corresponding to the line segment L3, on the plane work area of the work memory 2A are inverted. As a result, when the bit inversion is completed, the triangle which is filled inside as shown in Fig. 23 is achieved. But, it is an incompletely-filled triangle with some of the line segments on the right side lacking by bit (pixel).

Subsequently, as shown in Fig. 24, only the line segments L2 and L3 (represented by hatched squares in Fig. 24) whose vectors are in the upward direction (which makes an upward vertical-coordinate variation) among the line segments L1 to L3 which are obtained based on the outline information are further drawn on the work memory 2A, to achieve a properly-filled triangle (Step S23 of Fig. 6).

After that, as shown in Fig. 25, the drawing information on the plane work area of the work memory 2A is transferred with extension to the plane display area of the display memory 4 through the bus 10 and the display circuit 3 (Step S24 of Fig. 6).

As a result, the triangle which is properly filled inside is displayed on the CRT 5 by the display circuit 3 based on the information stored in the display memory 4.

Thus, in the method for filling a polygon of the second preferred embodiment, for a polygon without self-intersection, some of the line segments constituting the outline of the polygon whose vector are in the upward direction (which makes an upward vertical-coordinate variation) are further drawn on the work memory where the incompletely-filled polygon is drawn by filling the polygon with one of the line segments constituting the outline of the polygon selected sequentially in the counterclockwise rotation, to properly fill the polygon, and the drawing information is transferred with extension from the work memory to the display memory.

As the amount of memories needed for one pixel is smaller in the work memory than in the display memory, drawing some of the line segments whose vectors are in the upward direction on the work memory as revealed in the second preferred embodiment reduces the number of memory accesses and suppresses the load on the CPU 1, thus achieving faster drawing of filled polygon.

Moreover, instead of all the line segments constituting the polygon, only the line segments on the right side (whose vectors are in the upward direction) of the polygon which are needed for proper filling of the polygon, are efficiently drawn, and much faster drawing of filled polygon is thereby realized.

Further, though the bits inclusive of those for the line segments are inverted (in a first inversion mode) in the above method for filling a polygon of the second preferred embodiment, the bits may be inverted non inclusive of those for the line segments (in a second inversion mode). When the bits non inclusive of those for the line segments are inverted, some of the line segments on the left side of the polygon lack. Therefore, after the bit inversion, only the line segments on the left side (whose vectors are in the downward direction) of the polygon are drawn.

Furthermore, though the bits in the right direction of the line segments constituting the polygon are inverted in the above method for filling a polygon of the second preferred embodiment, the bits in the left direction of the line segments constituting the polygon may be inverted, based on the same principle. In this case, by further drawing the line segments whose vectors are in the downward direction, the properly-filled polygon can be surely achieved from the incompletely-filled polygon with some of the line segments on the left side lacking by bit (pixel).

The Third Preferred Embodiment

Figs. 26 to 31 are illustrations of the method for filling a polygon in accordance with the third preferred embodiment. Any of the hardware constructions of Figs. 1 to 4 may be used to implement the method for filling a polygon of the third preferred embodiment, and for explanatory convenience, the method using the hardware construction of Fig. 1 will be discussed below.

First, the CPU 1 reads a program recording a method for filling a polygon of the third preferred embodiment from the CD-ROM 6 through the bus 10. The method for filling a polygon of the third preferred embodiment is used for processing a polygon without self-intersection.

Then, the coordinates of three vertices {(P1 (H8, S2), P2 (H2, S15), P3 (H16, S11)} are externally obtained as the outline information of the polygon to be filled having the outline of Fig. 26 and are stored in a predetermined area of the memory 2 (Step S31 of Fig. 7). The plane work area of the work memory 2A in the memory 2 is blank in an initial state.

In the method for filling a polygon of the third preferred embodiment, as shown in Fig. 14, for the triangle to be filled, the line segments L1, L2 and L3 are selected in the order of the vertices P1, P2 and P3 in the counterclockwise rotation like in the first and second preferred embodiments. The following operation is performed under the control of the CPU 1 based on the program read from the CD-ROM 6.

Next, the bit inversion is performed on the work memory 2A as shown in Figs. 27 to 29 (Step S32 of Fig. 7). The bit inversion will be discussed below.

First, as shown in Fig. 27, the line segment L1 is obtained from the coordinates of the vertices P1 and P2 given as the outline information, using an already-existing linear algorithm such as Bresenham algorithm, and all the bits in the right direction (in a predetermined horizontal direction) of the line segment L1 on the horizontal coordinate, inclusive of those corresponding to the line segment L1, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw.

Secondly, as shown in Fig. 28, the line segment L2 is obtained from the coordinates of the vertices P2 and P3 given as the outline information, and all the bits in the right direction of the line segment L2 on the horizontal coordinate, inclusive of those corresponding to the line segment L2, on the plane work area of the work memory 2A are inverted. As a result, some of the already-drawn bits of Fig. 27 are inverted again to be back in blank.

Similarly, the line segment L3 is obtained based on the outline information, and all the bits in the right direction of the line segment L3 on the horizontal coordinate inclusive of those corresponding to the line segment L3, on the plane work area of the work memory 2A are inverted. As a result, when the bit inversion is completed, the triangle filled inside as shown in Fig. 29 is achieved. But, it is an incompletely-filled triangle with some of the line segments on the right side lacking by bit (pixel).

After that, as shown in Fig. 30, the drawing information on the plane work area of the work memory 2A is transferred with extension to the plane display area of the display memory 4 through the bus 10 and the display circuit 3 (Step S33 of Fig. 7).

Subsequently, as shown in Fig. 31, only the line segments L2 and L3 (represented by hatched squares in Fig. 31) whose vectors are in the upward direction (which makes an upward vertical-coordinate variation) among the line segments L1 to L3 which are obtained based on the outline information are further drawn on the work memory 2A, to achieve a properly-filled triangle (Step S34 of Fig. 7).

As a result, the triangle which is properly filled inside is displayed on the CRT 5 by the display circuit 3 based on the information stored in the display memory 4.

Thus, in the method for filling a polygon of the third preferred embodiment, for a polygon without self-intersection, some of the line segments constituting the outline of the polygon whose vectors are in the upward direction (which makes an upward vertical-coordinate variation) are further drawn on the display memory 4 where the incompletely-filled polygon is drawn by filling the polygon with one of the line segments constituting the outline of the polygon selected sequentially in the counterclockwise rotation, to properly fill the polygon.

Therefore, instead of all the line segments constituting the polygon, only the line segments on the right side (whose vectors are in the upward direction) of the polygon which are needed for proper filling of the polygon, are efficiently drawn, and much faster drawing of filled polygon is thereby realized like in the second preferred embodiment.

Further, though the bits inclusive of those for the line segments are inverted (in a first inversion mode) in the above method for filling a polygon of the third preferred embodiment, the bits may be inverted non inclusive of those for the line segments (in a second inversion mode). When the bits non inclusive of those for the line segments are inverted, some of the line segments on the left side of the polygon lack. Therefore, after the bit inversion, only the line segments on the left side (whose vectors are in the downward direction) of the polygon are drawn.

Furthermore, though the bits in the right direction of the line segments constituting the polygon are inverted in the above method for filling a polygon of the third preferred embodiment, the bits in the left direction of the line segments constituting the polygon may be inverted, based on the same principle. In this case, by further drawing the line segments whose vectors are in the downward direction, the properly-filled polygon can be surely achieved from the incompletely-filled polygon with some of the line segments on the left side lacking by bit (pixel).

The Fourth Preferred Embodiment

Figs. 32 to 38 are illustrations of the method for filling a polygon in accordance with the fourth preferred embodiment. Any of the hardware constructions of Figs. 1 to 4 may be used to implement the method for filling a polygon of the fourth preferred embodiment, and for explanatory convenience, the method using the hardware construction of Fig. 1 will be discussed below.

First, the CPU 1 reads a program recording a method for filling a polygon of the fourth preferred embodiment from the CD-ROM 6 through the bus 10. The method for filling a polygon of the fourth preferred embodiment is used for processing a polygon without self-intersection.

Then, the coordinates of three vertices {(P1 (H8, S2), P2 (H2, S15), P3 (H16, S11)} are externally obtained as the outline information of the polygon to be filled having the outline of Fig. 32 and are stored in a predetermined area of the memory 2 (Step S41 of Fig. 8). The plane work area of the work memory 2A in the memory 2 is blank in an initial state.

In the method for filling a polygon of the fourth preferred embodiment, as shown in Fig. 33, for the triangle to be filled, the line segments L11 (H16-h8, S11-S2), L12 (H2-H16, S15-S11) and L13 (H8-H2, S2-S15) are selected in the order of the vertices P1, P3 and P2 in the clockwise rotation. The following operation is performed under the control of the CPU 1 based on the program read from the CD-ROM 6.

Next, the bit inversion is performed on the work memory 2A as shown in Figs. 34 to 36 (Step S42 of Fig. 8). The bit inversion will be discussed below. First, as shown in Fig. 34, the line segment L11 is obtained from the coordinates of the vertices P1 and P3 given as the outline information, using an already-existing linear algorithm such as Bresenham algorithm, and all the bits in the right direction (in a predetermined horizontal direction) of the line segment L11 on the horizontal coordinate, inclusive of those corresponding to the line segment L11, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw. Fig 34 shows a halfway state of the bit inversion of the line segment L11.

Secondly, the line segment L12 is obtained from the coordinates of the vertices P2 and P3 given as the outline information and all the bits in the right direction of the line segment L12 on the horizontal coordinate, inclusive of those corresponding to the line segment L12, on the plane work area of the work memory 2A are inverted.

Similarly, as shown in Fig. 35, the line segment L13 is obtained based on the outline information, and all the bits in the right direction of the line segment L13 on the horizontal coordinate, inclusive of those corresponding to the line segment L13, on the plane work area of the work memory 2A are inverted. At this time, some of the bits which are already drawn by the bit inversion based on the line segments L11 and L12 are inverted again to be back in blank. Fig. 35 shows a halfway state of the bit inversion of the line segment L13.

When the bit inversion is completed, the triangle which is filled inside as shown in Fig. 36 is achieved. But, it is an incompletely-filled triangle with some of the line segments on the right side lacking by bit (pixel).

Subsequently, as shown in Fig. 37, only the line segments L11 and L12 (represented by hatched squares in Fig. 37) whose vectors are in the downward direction (which makes a downward vertical-coordinate variation) among the line segments L11 to L13 which are obtained based on the outline information are further drawn on the work memory 2A, to achieve a properly-filled triangle (Step S43 of Fig. 8).

After that, as shown in Fig. 38, the drawing information on the plane work area of the work memory 2A is transferred with extension to the plane display area of the display memory 4 through the bus 10 and the display circuit 3 (Step S44 of Fig. 8).

As a result, the triangle which is properly filled inside is displayed on the CRT 5 by the display circuit 3 based on the information stored in the display memory 4.

Thus, in the method for filling a polygon of the fourth preferred embodiment, for a polygon without self-intersection, some of the line segments constituting the outline of the polygon whose vectors are in the downward direction (which makes a downward vertical-coordinate variation) are further drawn on the work memory where the incompletely-filled polygon is drawn by filling the polygon with one of the line segments constituting the outline of the polygon selected sequentially in the clockwise rotation, to properly fill the polygon, and the drawing information is transferred with extension from the work memory to the display memory.

As the amount of memories needed for one pixel is smaller in the work memory than in the display memory, drawing some of the line segments whose vectors are in the downward direction on the work memory as revealed in the fourth preferred embodiment reduces the number of memory accesses and suppresses the load on the CPU 1, thus achieving faster drawing of filled polygon.

Moreover, instead of all the line segments constituting the polygon, only the line segments on the right side (whose vectors are in the downward direction) of the polygon which are needed for proper filling of the polygon, are efficiently drawn, and much faster drawing of filled polygon is thereby realized.

Further, though the bits inclusive of those for the line segments are inverted (in a first inversion mode) in the above method for filling a polygon of the fourth preferred embodiment, the bits may be inverted non inclusive of those for the line segments (in a second inversion mode). When the bits non inclusive of those for the line segments are inverted, some of the line segments on the left side of the polygon lack. Therefore, after the bit inversion, only the line segments on the left side (whose vectors are in the upward direction) of the polygon are drawn.

Furthermore, though the bits in the right direction of the line segments constituting the polygon are inverted in the above method for filling a polygon of the fourth preferred embodiment, the bits in the left direction of the line segments constituting the polygon may be inverted, based on the same principle. In this case, by further drawing the line segments whose vectors are in the upward direction, the properly-filled polygon can be surely achieved from the incompletely-filled polygon with some of the line segments on the left side lacking by bit (pixel).

The Fifth Preferred Embodiment

Figs. 39 to 44 are illustrations of the method for filling a polygon in accordance with the fifth preferred embodiment. Any of the hardware constructions of Figs. 1 to 4 may be used to implement the method for filling a polygon of the fifth preferred embodiment, and for explanatory convenience, the method using the hardware construction of Fig. 1 will be discussed below.

First, the CPU 1 reads a program recording a method for filling a polygon of the fifth preferred embodiment from the CD-ROM 6 through the bus 10. The method for filling a polygon of the fifth preferred embodiment is used for processing a polygon without self-intersection.

Then, the coordinates of three vertices {(P1 (H8, S2), P2 (H2, S15), P3 (H16, S11)} are externally obtained as the outline information of the polygon to be filled having the outline of Fig. 39 and are stored in a predetermined area of the memory 2 (Step S51 of Fig. 9). The plane work area of the work memory 2A in the memory 2 is blank in an initial state.

In the method for filling a polygon of the fifth preferred embodiment, as shown in Fig. 33, for the triangle to be filled, the line segments L11, L12 and L13 are selected in the order of the vertices P1, P3 and P2 in the clockwise rotation like in the fourth preferred embodiment. The following operation is performed under the control of the CPU 1 based on the program read from the CD-ROM 6.

Next, the bit inversion is performed on the work memory 2A as shown in Figs. 40 to 42 (Step S52 of Fig. 9). The bit inversion will be discussed below.

First, as shown in Fig. 40, the line segment L11 is obtained from the coordinates of the vertices P1 and P3 given as the outline information, using an already-existing linear algorithm such as Bresenham algorithm, and all the bits in the right direction (in a predetermined horizontal direction) of the line segment L11 on the horizontal coordinate, inclusive of those corresponding to the line segment L11, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw. Fig. 40 shows a halfway state of the bit inversion of the line segment L11.

Secondly, the line segment L12 is obtained from the coordinates of the vertices P2 and P3 given as the outline information and all the bits in the right direction of the line segment L12 on the horizontal coordinate, inclusive of those corresponding to the line segment L12, on the plane work area of the work memory 2A are inverted.

Similarly, as shown in Fig. 41, the line segment L13 is obtained based on the outline information, and all the bits in the right direction of the line segment L13 on the horizontal coordinate, inclusive of those corresponding to the line segment L13, on the plane work area of the work memory 2A are inverted. At this time, some of the bits which are already drawn by the bit inversion based on the line segments L11 and L12 are inverted again to be back in blank. Fig. 41 shows a halfway state of the bit inversion of the line segment L13.

When the bit inversion is completed, the triangle which is filled inside as shown in Fig. 36 is achieved. But, it is an incompletely-filled triangle with some of the line segments on the right side lacking by bit (pixel).

After that, as shown in Fig. 43, the drawing information on the plane work area of the work memory 2A is transferred with extension to the plane display area of the display memory 4 through the bus 10 and the display circuit 3 (Step S53 of Fig. 9).

Subsequently, as shown in Fig. 44, only the line segments L11 and L12 (represented by hatched squares in Fig. 43) whose vectors are in the downward direction (which makes a downward vertical-coordinate variation) among the line segments L11 to L13 which are obtained based on the outline information are further drawn on the display memory 4, to achieve a properly-filled triangle (Step S54 of Fig. 9).

As a result, the triangle which is properly filled inside is displayed on the CRT 5 by the display circuit 3 based on the information stored in the display memory 4.

Thus, in the method for filling a polygon of the fifth preferred embodiment, for a polygon without self-intersection, some of the line segments constituting the outline of the polygon whose vectors are in the downward direction (which makes a downward vertical-coordinate variation) are further drawn on the display memory where the incompletely-filled polygon is drawn by filling the polygon with one of the line segments constituting the outline of the polygon selected sequentially in the clockwise rotation, to properly fill the polygon.

Therefore, instead of all the line segments constituting the polygon, only the line segments on the right side (whose vectors are in the downward direction) of the polygon which are needed for proper filling of the polygon, are efficiently drawn, and much faster drawing of filled polygon is thereby realized.

Further, though the bits inclusive of those for the line segments are inverted (in a first inversion mode) in the above method for filling a polygon of the fifth preferred embodiment, the bits may be inverted non inclusive of those for the line segments (in a second inversion mode). When the bits non inclusive of those for the line segments are inverted, some of the line segments on the left side of the polygon lack. Therefore, after the bit inversion, only the line segments on the left side (whose vectors are in the upward direction) of the polygon are drawn.

Furthermore, though the bits in the right direction of the line segments constituting the polygon are inverted in the above method for filling a polygon of the fifth preferred embodiment, the bits in the left direction of the line segments constituting the polygon may be inverted, based on the same principle. In this case, by further drawing the line segments whose vectors are in the upward direction, the properly-filled polygon can be surely achieved from the incompletely-filled polygon with some of the line segments on the left side lacking by bit (pixel).

The Sixth Preferred Embodiment

Figs. 45 to 51 are illustrations of the method for filling a polygon in accordance with the sixth preferred embodiment. Any of the hardware constructions of Figs. 1 to 4 may be used to implement the method for filling a polygon of the sixth preferred embodiment, and for explanatory convenience, the method using the hardware construction of Fig. 1 will be discussed below.

First, the CPU 1 reads a program recording a method for filling a polygon of the sixth preferred embodiment from the CD-ROM 6 through the bus 10. The method for filling a polygon of the sixth preferred embodiment is used for processing a polygon without self-intersection.

Then, the coordinates of three vertices {(P1 (H8, S2), P2 (H2, S15), P3 (H16, S11)} are externally obtained as the outline information of the polygon to be filled having the outline of Fig. 45 and are stored in a predetermined area of the memory 2 (Step S61 of Fig. 10). The plane work area of the work memory 2A in the memory 2 is blank in an initial state.

In the method for filling a polygon of the sixth preferred embodiment, as shown in Fig. 14, for the triangle to be filled, the line segments L1, L2 and L3 are selected in the order of the vertices P1, P2 and P3 in the counterclockwise rotation like in the first to third preferred embodiments. The following operation is performed under the control of the CPU 1 based on the program read from the CD-ROM 6.

Next, the bit inversion is performed on the work memory 2A as shown in Figs. 46 to 50 (Step S62 of Fig. 10). The bit inversion will be discussed below.

First, the line segment L1 is obtained from the coordinates of the vertices P1 and P2 given as the outline information, using an already-existing linear algorithm such as Bresenham algorithm, and as the line segment L1 is in the downward direction, all the bits in the right direction (in a predetermined horizontal direction) of the line segment L1 on the horizontal coordinate, inclusive of those corresponding to the line segment L1, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw (the bit inversion in a first inversion mode), as shown in Fig. 46.

Secondly, the line segment L2 is obtained from the coordinates of the vertices P2 and P3 given as the outline information and as the line segment L2 is in the upward direction, all the bits in the right direction (in a predetermined horizontal direction) of the line segment L2 on the horizontal coordinate, not inclusive of those corresponding to the line segment L2, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw (the bit inversion in a second inversion mode, an area AR1 consisting of the solid squares in Fig. 48 is to be inverted), as shown in Fig. 47. As a result, some of the already-drawn bits of Fig. 46 are inverted again to be back in blank.

Similarly, the line segment L3 is obtained based on the outline information, and as the line segment L3 is in the upward direction, all the bits in the right direction of the line segment L3 on the horizontal coordinate, not inclusive of those for the line segment L3, on the plane work area of the work memory 2A are inverted (the bit inversion in a second inversion mode, an area AR2 consisting of the solid squares in Fig. 49 is to be inverted). As a result, when the bit inversion is completed, the triangle which is filled inside as shown in Fig. 50 is achieved.

After that, as shown in Fig. 51, the drawing information on the plane work area of the work memory 2A is transferred with extension to the plane display area of the display memory 4 through the bus 10 and the display circuit 3 (Step S63 of Fig. 10).

As a result, the triangle which is properly filled inside is displayed on the CRT 5 by the display circuit 3 based on the information stored in the display memory 4.

Thus, in the method for filling a polygon of the sixth preferred embodiment, for a polygon without self-intersection, with one of the line segments constituting the outline of the polygon selected sequentially in the counterclockwise rotation, the polygon is filled to draw a filled polygon on the work memory. At this time, if the line segment is in the upward direction, the bit inversion is performed in the second inversion mode, not inclusive of bits for the line segment, and otherwise (the line segment is in the downward direction or in the horizontal direction), the bit inversion is performed in the first inversion mode, inclusive of bits for the line segment.

As the amount of memories needed for one pixel is smaller in the work memory than in the display memory, performing all the steps for filling the polygon on the work memory as revealed in the sixth preferred embodiment reduces the number of memory accesses and suppresses the load on the CPU 1, thus achieving faster drawing of filled polygon.

Moreover, since a complete filling of a polygon can be achieved by only one kind of operation, i.e., the bit inversion in the first inversion mode or the second inversion mode, much faster drawing of filled polygon is realized.

Furthermore, though the bits in the right direction of the line segments constituting the polygon are inverted in the above method for filling a polygon of the sixth preferred embodiment, the bits in the left direction of the line segments constituting the polygon may be inverted, based on the same principle. In this case, when the vector of the selected one of the line segments constituting the outline of the polygon is in the downward direction, the bits not inclusive of those for the line segment are inverted, and otherwise the bits inclusive of those for the line segment are inverted.

The Seventh Preferred Embodiment

Figs. 52 to 58 are illustrations of the method for filling a polygon in accordance with the seventh preferred embodiment. Any of the hardware constructions of Figs. 1 to 4 may be used to implement the method for filling a polygon of the seventh preferred embodiment, and for explanatory convenience, the method using the hardware construction of Fig. 1 will be discussed below.

First, the CPU 1 reads a program recording a method for filling a polygon of the seventh preferred embodiment from the CD-ROM 6 through the bus 10. The method for filling a polygon of the seventh preferred embodiment is used for processing a polygon without self-intersection.

Then, the coordinates of three vertices {(P1 (H8, S2), P2 (H2, S15), P3 (H16, S11)} are externally obtained as the outline information of the polygon to be filled having the outline of Fig. 52 and are stored in a predetermined area of the memory 2 (Step S71 of Fig. 11). The plane work area of the work memory 2A in the memory 2 is blank in an initial state.

In the method for filling a polygon of the seventh preferred embodiment, as shown in Fig. 33, for the triangle to be filled, the line segments L11, L12 and L13 are selected in the order of the vertices P1, P3 and P2 in the clockwise rotation like in the fourth and fifth preferred embodiments. The following operation is performed under the control of the CPU 1 based on the program read from the CD-ROM 6.

Next, the bit inversion is performed on the work memory 2A as shown in Figs. 53 to 57 (Step S72 of Fig. 11). The bit inversion will be discussed below.

First, the line segment L11 is obtained from the coordinates of the vertices P1 and P3 given as the outline information, using an already-existing linear algorithm such as Bresenham algorithm, and as the line segment L11 is in the downward direction, all the bits in the right direction (in a predetermined horizontal direction) of the line segment L11 on the horizontal coordinate, not inclusive of those corresponding to the line segment L11, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw (the bit inversion in a second inversion mode, an area AR3 consisting of the solid squares in Fig. 54 is to be inverted), as shown in Fig. 53.

Secondly, the line segment L12 is obtained from the coordinates of the vertices P2 and P3 given as the outline information and as the line segment L12 is in the downward direction, all the bits in the right direction of the line segment L12 on the horizontal coordinate, not inclusive of those corresponding to the line segment L12, on the plane work area (1 bit/pixel) of the work memory 2A are inverted to draw (the bit inversion in a second inversion mode, an area AR4 consisting of the solid squares in Fig. 55 is to be inverted). As a result, some of the already-drawn bits of Fig. 53 are inverted again to be back in blank.

Similarly, the line segment L13 is obtained based on the outline information, and as the line segment L13 is in the upward direction, all the bits in the right direction of the line segment L13 on the horizontal coordinate, inclusive of those corresponding to the line segment L13, on the plane work area of the work memory 2A are inverted (the bit inversion in a first inversion mode), as shown in Fig. 56. Fig. 56 shows a halfway state of the bit inversion of the line segment L13.

As a result, when the bit inversion is completed, the triangle which is filled inside as shown in Fig. 57 is achieved.

After that, as shown in Fig. 58, the drawing information on the plane work area of the work memory 2A is transferred with extension to the plane display area of the display memory 4 through the bus 10 and the display circuit 3 (Step S73 of Fig. 11).

As a result, the triangle which is properly filled inside is displayed on the CRT 5 by the display circuit 3 based on the information stored in the display memory 4.

Thus, in the method for filling a polygon of the seventh preferred embodiment, for a polygon without self-intersection, with one of the line segments constituting the outline of the polygon selected sequentially in the clockwise rotation, the polygon is filled to draw a filled polygon on the work memory. At this time, if the line segment is in the downward direction, the bit inversion is performed in the second inversion mode, not inclusive of bits for the line segment, and otherwise (the line segment is in the upward direction or in the horizontal direction), the bit inversion is performed in the first inversion mode, inclusive of bits for the line segment.

As the amount of memories needed for one pixel is smaller in the work memory than in the display memory, performing all the steps for filling the polygon on the work memory as revealed in the seventh preferred embodiment reduces the number of memory accesses and suppresses the load on the CPU 1, thus achieving faster drawing of filled polygon.

Moreover, since a complete filling of a polygon can be achieved by only one kind of operation, i.e., the bit inversion in the first inversion mode or the second inversion mode, much faster drawing of filled polygon is realized.

Furthermore, though the bits in the right direction of the line segments constituting the polygon are inverted in the above method for filling a polygon of the seventh preferred embodiment, the bits in the left direction of the line segments constituting the polygon may be inverted, based on the same principle. In this case, when the vector of the selected one of the line segments constituting the outline of the polygon is in the upward direction, the bits not inclusive of those for the line segment are inverted, and otherwise the bits inclusive of those for the line segment are inverted.

While the invention has been shown and described in detail, the foregoing description is in all aspects illustrative and not restrictive. It is therefore understood that numerous modifications and variations can be devised without departing from the scope of the invention.

For a polygon without self-intersection, with one of line segments constituting the outline of the polygon selected sequentially in the counterclockwise rotation, all the bits in the right direction of the selected line segment on the horizontal coordinate, inclusive of those for the selected line segment, on the plane work area (1 bit/pixel) of the work memory are inverted to draw (Step S22). Subsequently, only some of the line segments constituting the outline of the polygon whose vectors are in the upward direction (which makes an upward vertical-coordinate variation) are further drawn on the work memory (2A) (Step S23). Finally, the drawing information is transferred with extension from the work memory to the display memory (Step S24). Having the above steps, a method for achieving proper filling of a polygon with a decrease in load on a processor is provided.

Claims

1. A method for filling a polygon, using an image processing system which has a work memory (2A, 7A) having a rectangular memory area of a plurality of bits, for storing information of one pixel by n ($n \geqq 1$) bits and a display memory (4, 7B) having a rectangular memory area of a plurality of bits, for storing information of one pixel by m ($m > n$) bits and displays an image based on said information stored in said display memory, to fill the inside of said polygon, said work memory being blank in an initial state,

    said method comprising the steps of:

    (a) obtaining outline information defining line segments constituting an outline of a polygon to be filled ⟨S11, S21, S41⟩;

(b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of said work memory corresponding to each of said line segments while selecting each of said line segments constituting said outline of said polygon to be filled in a predetermined rotation based on said outline information ⟨S12, S22, S42⟩;

(c) further drawing said line segments constituting said outline of said polygon to be filled on said work memory where said drawing of filled polygon is performed based on said outline information ⟨S13, S23, S43⟩; and

(d) transferring information stored in said work memory to said display memory after said step (c) ⟨S14, S24, S44⟩.

2. The method of claim 1, wherein

said step (c) includes the step ⟨S13⟩ of drawing all said line segments constituting said outline of said polygon to be filled and selected in said predetermined rotation.

3. The method of claim 1, wherein

said polygon to be filled includes a polygon without self-intersection,

said bit inversion ⟨S22, S42⟩ in said step (b) includes one of a first inversion mode for inverting bits inclusive of those corresponding to said line segments on said work memory and a second inversion mode for inverting bits not inclusive of those corresponding to said line segments,

said step (c) includes the step ⟨S23, S43⟩ of drawing only some of said line segments constituting said outline of said polygon to be filled and selected in said predetermined rotation whose vectors have a vertical variation in a predetermined vertical direction, said predetermined vertical direction being determined by said bit inversion, said predetermined rotation and said predetermined horizontal direction, in which some of said line segments which fail to be drawn, lacking, in said bit inversion of the step (b) have a vertical variation.

4. A method for filling a polygon, using an image processing system which has a work memory ⟨2A, 7A⟩ having a rectangular memory area of a plurality of bits, for storing information of one pixel by n (n ≧ 1) bits and a display memory ⟨4, 7B⟩ having a rectangular memory area of a plurality of bits, for storing information of one pixel by m (m > n) bits and displays an image based on said information stored in said display memory, to fill the inside of said polygon, said work memory being blank in an initial state,

said method comprising the steps of:

(a) obtaining outline information defining line segments constituting an outline of a polygon without self-intersection to be filled ⟨S31, S51⟩; and

(b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of said work memory corresponding to each of said line segments while selecting each of said line segments constituting said outline of said polygon to be filled in a predetermined rotation based on said outline information ⟨S32, S52⟩;

wherein said bit inversion includes one of a first inversion mode for inverting bits inclusive of those corresponding to said line segments on said work memory and a second inversion mode for inverting bits not inclusive of those corresponding to said line segments,

said method further comprising the steps of;

(c) transferring information stored in said work memory to said display memory ⟨S33, S53⟩ after said step (b); and

(d) drawing only some of said line segments constituting said outline of said polygon to be filled and selected in said predetermined rotation whose vectors have a vertical variation in a predetermined vertical direction on said display memory based on said outline information ⟨S34, S54⟩ after said step (c), said predetermined vertical direction being determined by said bit inversion, said predetermined rotation and said predetermined horizontal direction, in which some of said line segments which fail to be drawn, lacking, in said bit inversion of the step (b) have a vertical variation.

5. The method of claim 3 or 4, wherein

said bit inversion includes said first inversion mode,

said predetermined rotation includes a counterclockwise rotation,

said predetermined horizontal direction

includes a right direction, and

said predetermined vertical direction includes an upward direction.

6. The method of claim 3 or 4, wherein

said bit inversion includes said first inversion mode,

said predetermined rotation includes a clockwise rotation,

said predetermined horizontal direction includes a right direction, and

said predetermined vertical direction includes a downward direction.

7. The method of claim 3 or 4, wherein

said bit inversion includes said second inversion mode,

said predetermined rotation includes a counterclockwise rotation,

said predetermined horizontal direction includes a right direction, and

said predetermined vertical direction includes a downward direction.

8. The method of claim 3 or 4, wherein

said bit inversion includes said second inversion mode,

said predetermined rotation includes a clockwise rotation,

said predetermined horizontal direction includes a right direction, and

said predetermined vertical direction includes an upward direction.

9. The method of claim 3 or 4, wherein

said bit inversion includes said first inversion mode,

said predetermined rotation includes a counterclockwise rotation,

said predetermined horizontal direction includes a left direction, and

said predetermined vertical direction includes a downward direction.

10. The method of claim 3 or 4, wherein

said bit inversion includes said first inversion mode,

said predetermined rotation includes a clockwise rotation,

said predetermined horizontal direction includes a left direction, and

said predetermined vertical direction includes an upward direction.

11. A method for filling a polygon, using an image processing system which has a work memory (2A, 7A) having a rectangular memory area of a plurality of bits, for storing information of one pixel by n (n ≧ 1) bits and a display memory (4, 7B) having a rectangular memory area of a plurality of bits, for storing information of one pixel by m (m > n) bits and displays an image based on said information stored in said display memory, to fill the inside of said polygon, said work memory being blank in an initial state,

said method comprising the steps of:

(a) obtaining outline information defining line segments constituting an outline of a polygon without self-intersection to be filled (S61, S71);

(b) performing bit inversion to make drawing of filled polygon through inverting all bits in a predetermined horizontal direction on a horizontal coordinate of said work memory corresponding to each of said line segments while selecting each of said line segments constituting said outline of said polygon to be filled in a predetermined rotation based on said outline information (S62, S72);

(c) transferring information stored in said work memory to said display memory (S63, S73) after said step (b),

wherein said bit inversion includes a first inversion mode for inverting bits inclusive of those corresponding to said line segments on said work memory and a second inversion mode for inverting bits not inclusive of those corresponding to said line segments, and some of said line segment constituting said outline of said polygon to be filled and selected in said predetermined rotation whose vectors have a vertical variation in a first vertical direction are subjected to said bit inversion in said first inversion mode and some whose vectors have a vertical variation in a second vertical direction are subjected to said bit inversion in said second inversion mode, said first and second vertical directions being determined by said predetermined rotation and said predetermined horizontal direction so that said some of said line segments whose vectors have a vertical variation in said first vertical direction can be drawn with no lack when said bit inversion is performed in said first inversion mode and said some of said line segments whose vectors have a vertical variation in said second vertical direction can be drawn with no lack when said bit inversion is performed in said second inversion mode.

12. The method of claim 11, wherein

said predetermined rotation includes a counterclockwise rotation,
said predetermined horizontal direction includes a right direction,
said first predetermined vertical direction includes a downward direction,
and
said second predetermined vertical direction includes an upward direction.

13. The method of claim 11, wherein

said predetermined rotation includes a clockwise rotation,
said predetermined horizontal direction includes a right direction,
said first predetermined vertical direction includes an upward direction,
and
said second predetermined vertical direction includes a downward direction.

14. An image processing system used for the method for filling a polygon in accordance with claim 1.

15. An image processing system used for the method for filling a polygon in accordance with claim 4.

16. An image processing system used for the method for filling a polygon in accordance with claim 11.

FIG. 1

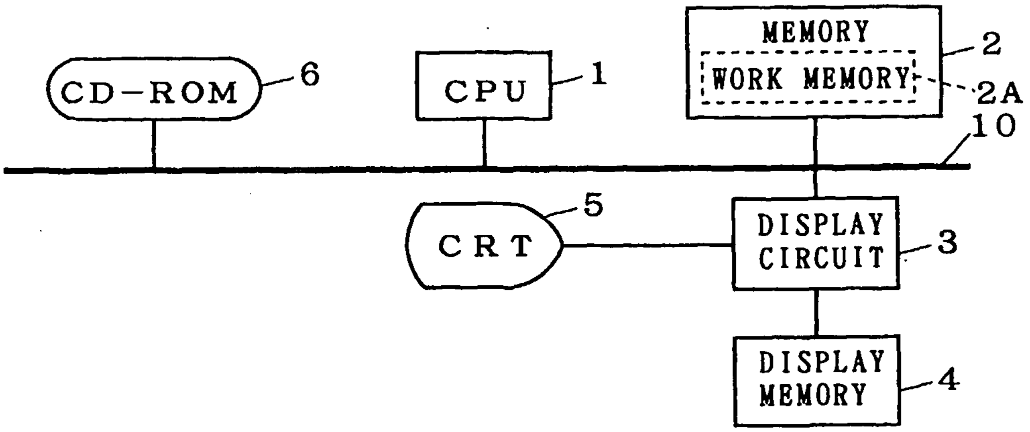

CD-ROM
6
CPU
1
MEMORY
WORK MEMORY
2
2A
10
CRT
5
DISPLAY CIRCUIT
3
DISPLAY MEMORY
4

FIG. 2

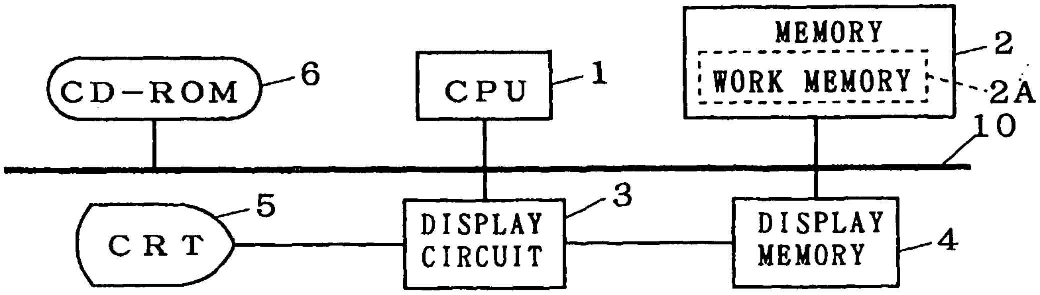

CD-ROM
6
CPU
1
MEMORY
WORK MEMORY
2
2A
10
CRT
5
DISPLAY CIRCUIT
3
DISPLAY MEMORY
4

FIG. 3

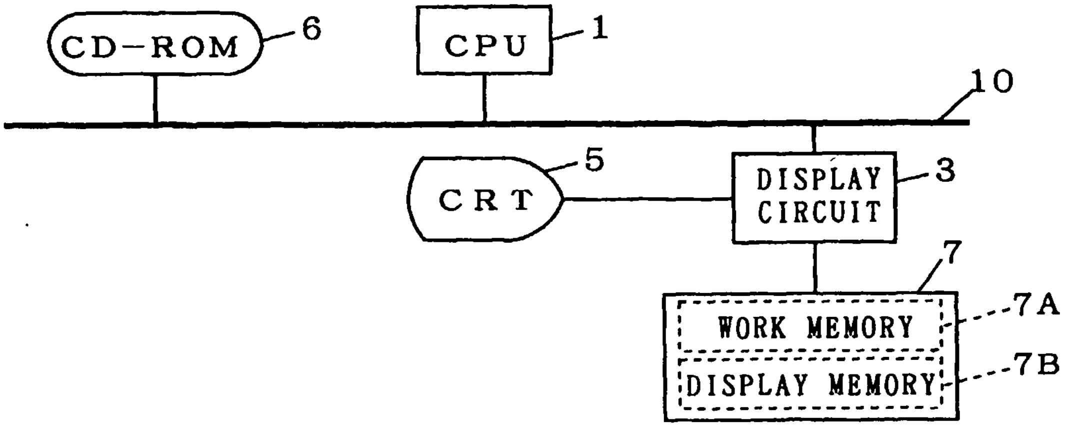

CD-ROM
6
CPU
1
10
CRT
5
DISPLAY CIRCUIT
3
7
WORK MEMORY
7A
DISPLAY MEMORY
7B

FIG. 4

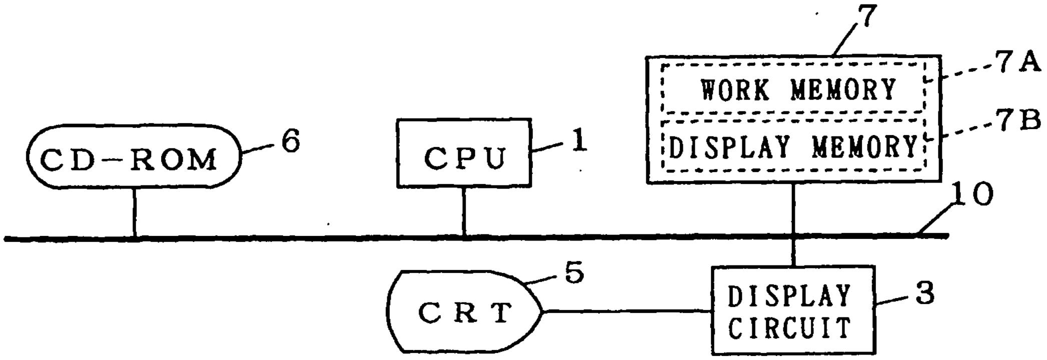

7
WORK MEMORY
7A
DISPLAY MEMORY
7B
CD-ROM
6
CPU
1
10
CRT
5
DISPLAY CIRCUIT
3

FIG. 5

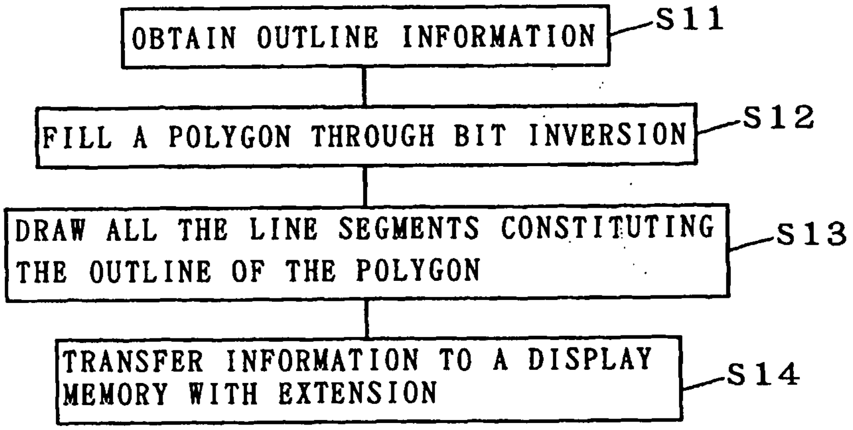

OBTAIN OUTLINE INFORMATION
S11
FILL A POLYGON THROUGH BIT INVERSION
S12
DRAW ALL THE LINE SEGMENTS CONSTITUTING THE OUTLINE OF THE POLYGON
S13
TRANSFER INFORMATION TO A DISPLAY MEMORY WITH EXTENSION
S14

FIG. 6

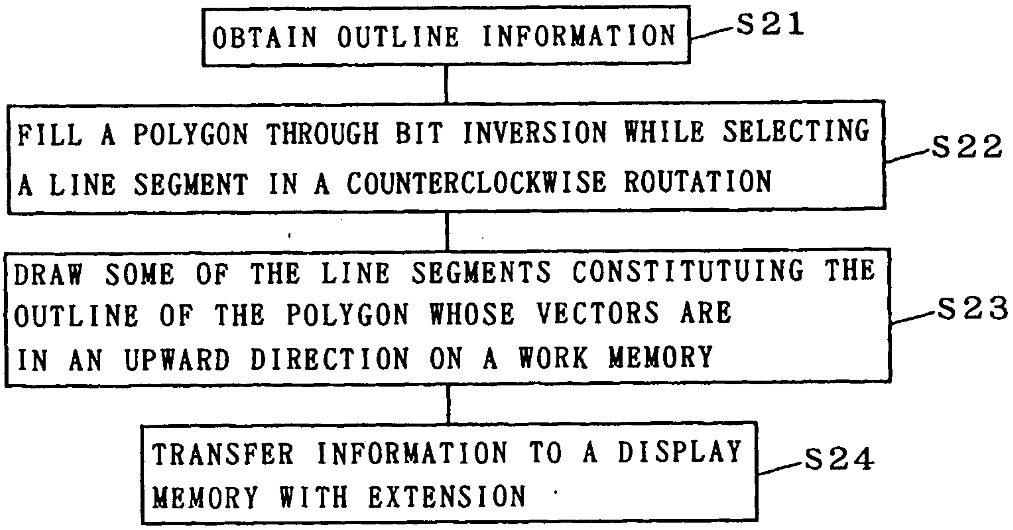

OBTAIN OUTLINE INFORMATION
S21
FILL A POLYGON THROUGH BIT INVERSION WHILE SELECTING A LINE SEGMENT IN A COUNTERCLOCKWISE ROUTATION
S22
DRAW SOME OF THE LINE SEGMENTS CONSTITUTUING THE OUTLINE OF THE POLYGON WHOSE VECTORS ARE IN AN UPWARD DIRECTION ON A WORK MEMORY
S23
TRANSFER INFORMATION TO A DISPLAY MEMORY WITH EXTENSION
S24

FIG. 7

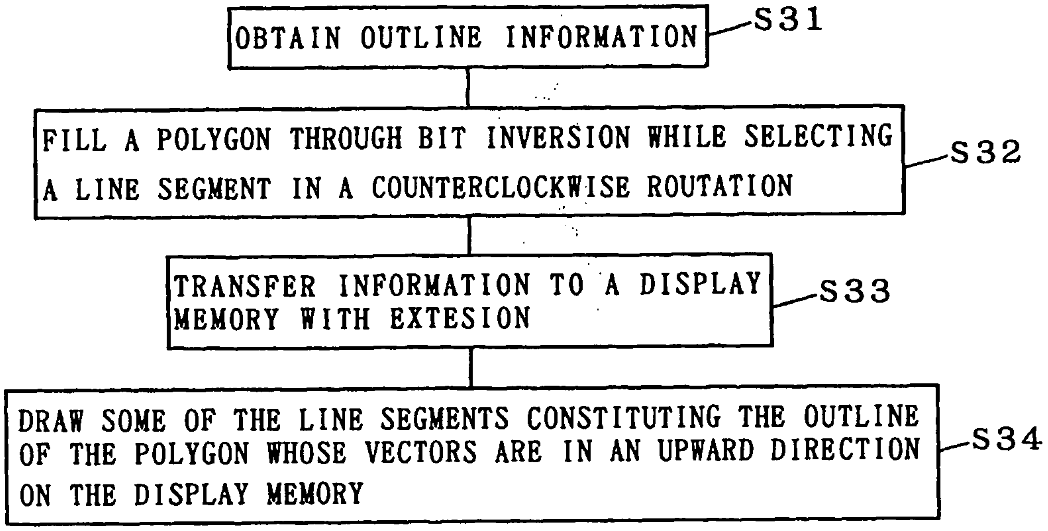

OBTAIN OUTLINE INFORMATION
S31
FILL A POLYGON THROUGH BIT INVERSION WHILE SELECTING A LINE SEGMENT IN A COUNTERCLOCKWISE ROUTATION
S32
TRANSFER INFORMATION TO A DISPLAY MEMORY WITH EXTESION
S33
DRAW SOME OF THE LINE SEGMENTS CONSTITUTING THE OUTLINE OF THE POLYGON WHOSE VECTORS ARE IN AN UPWARD DIRECTION ON THE DISPLAY MEMORY
S34

FIG. 8

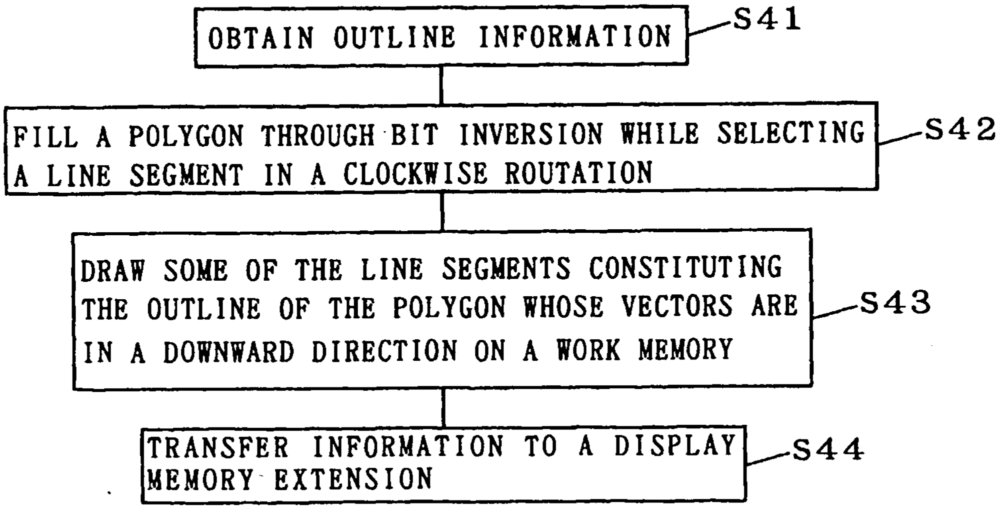

OBTAIN OUTLINE INFORMATION
S41
FILL A POLYGON THROUGH BIT INVERSION WHILE SELECTING A LINE SEGMENT IN A CLOCKWISE ROUTATION
S42
DRAW SOME OF THE LINE SEGMENTS CONSTITUTING THE OUTLINE OF THE POLYGON WHOSE VECTORS ARE IN A DOWNWARD DIRECTION ON A WORK MEMORY
S43
TRANSFER INFORMATION TO A DISPLAY MEMORY EXTENSION
S44

FIG. 9

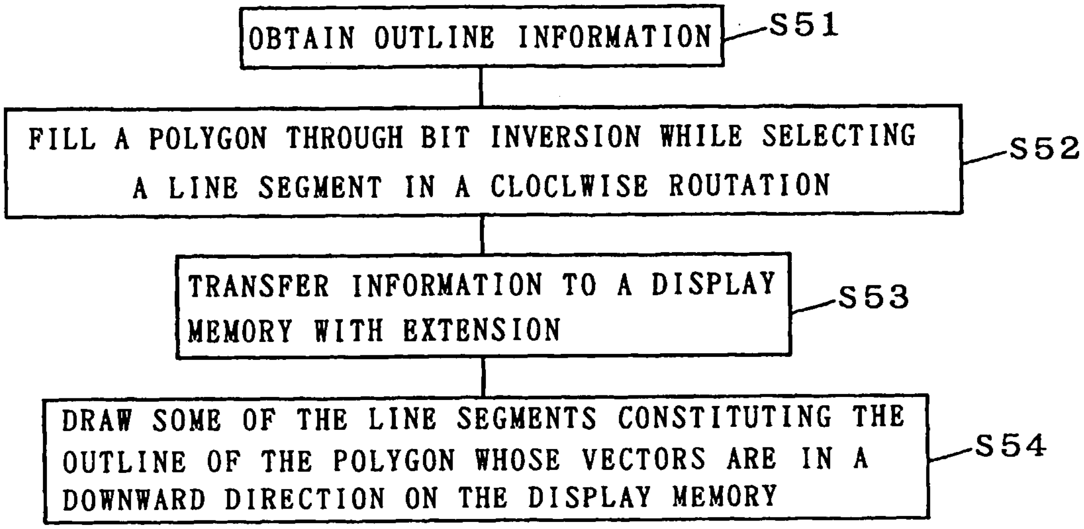

OBTAIN OUTLINE INFORMATION
S51
FILL A POLYGON THROUGH BIT INVERSION WHILE SELECTING A LINE SEGMENT IN A CLOCLWISE ROUTATION
S52
TRANSFER INFORMATION TO A DISPLAY MEMORY WITH EXTENSION
S53
DRAW SOME OF THE LINE SEGMENTS CONSTITUTING THE OUTLINE OF THE POLYGON WHOSE VECTORS ARE IN A DOWNWARD DIRECTION ON THE DISPLAY MEMORY
S54

FIG. 10

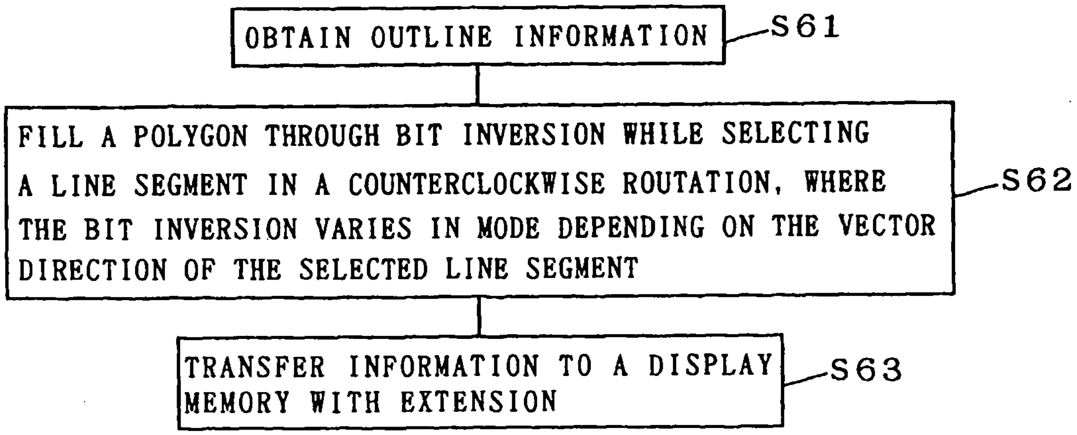
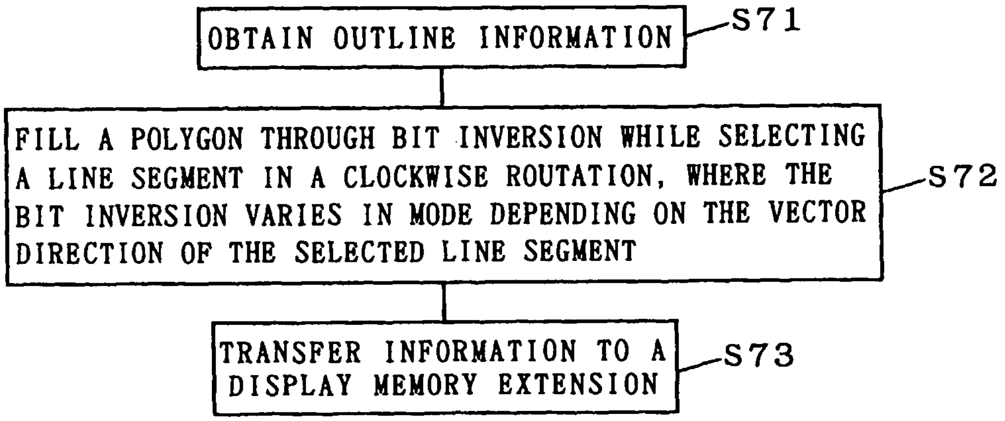

OBTAIN OUTLINE INFORMATION
S61
FILL A POLYGON THROUGH BIT INVERSION WHILE SELECTING A LINE SEGMENT IN A COUNTERCLOCKWISE ROUTATION, WHERE THE BIT INVERSION VARIES IN MODE DEPENDING ON THE VECTOR DIRECTION OF THE SELECTED LINE SEGMENT
S62
TRANSFER INFORMATION TO A DISPLAY MEMORY WITH EXTENSION
S63

FIG. 11



OBTAIN OUTLINE INFORMATION

FILL A POLYGON THROUGH BIT INVERSION WHILE SELECTING A LINE SEGMENT IN A CLOCKWISE ROUTATION, WHERE THE BIT INVERSION VARIES IN MODE DEPENDING ON THE VECTOR DIRECTION OF THE SELECTED LINE SEGMENT

TRANSFER INFORMATION TO A DISPLAY MEMORY EXTENSION

FIG. 12

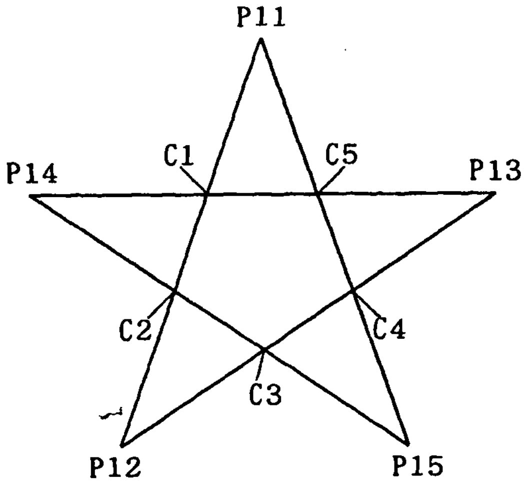
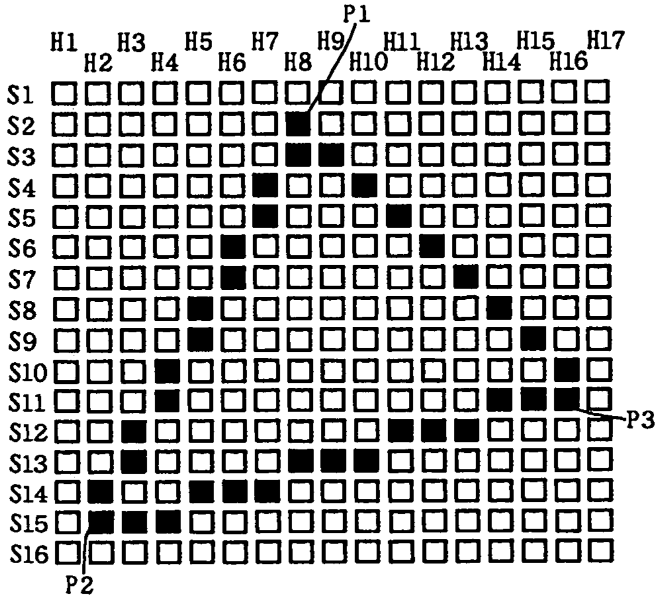


FIG. 13



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	□	□	■	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	□	□	□	■	□	□	□	□	□	□
S6	□	□	□	□	□	■	□	□	□	□	□	■	□	□	□	□	□
S7	□	□	□	□	□	■	□	□	□	□	□	□	■	□	□	□	□
S8	□	□	□	□	■	□	□	□	□	□	□	□	□	■	□	□	□
S9	□	□	□	□	■	□	□	□	□	□	□	□	□	□	■	□	□
S10	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	■	□
S11	□	□	□	■	□	□	□	□	□	□	□	□	□	■	■	■	□
S12	□	□	■	□	□	□	□	□	□	□	■	■	■	□	□	□	□
S13	□	□	■	□	□	□	□	■	■	■	□	□	□	□	□	□	□
S14	□	■	□	□	■	■	■	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 14

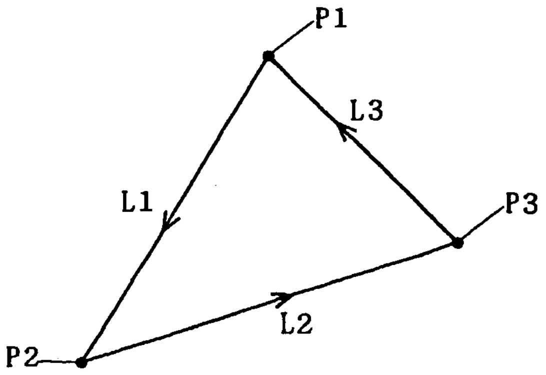
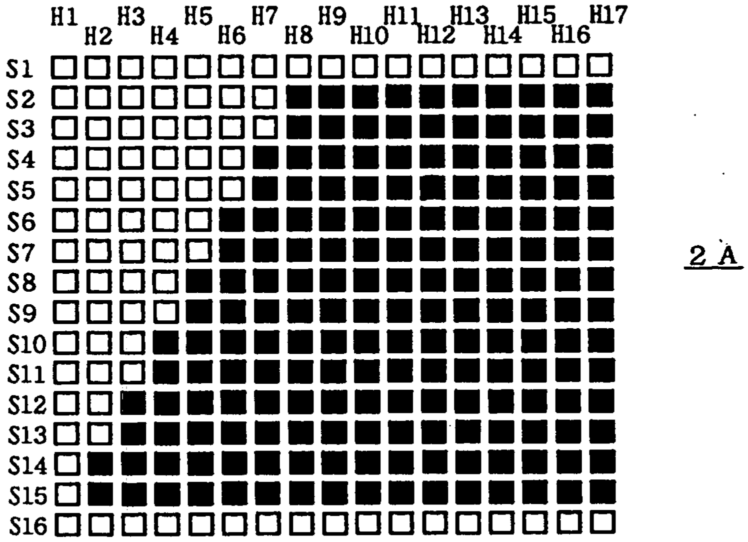
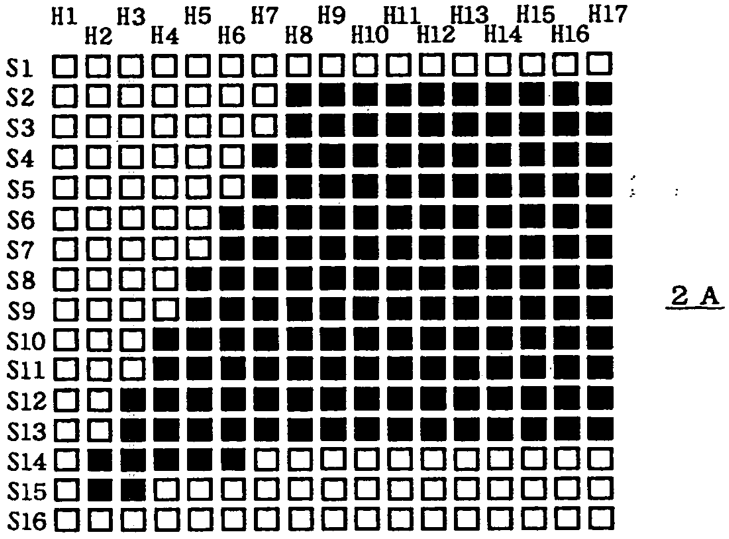


FIG. 15



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S15	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 16



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 17

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	□	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	□	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	□	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2 A

FIG. 18

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	▨	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	▨	▨	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	▨	■	■	▨	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	▨	■	■	■	▨	□	□	□	□	□	□
S6	□	□	□	□	□	▨	■	■	■	■	■	▨	□	□	□	□	□
S7	□	□	□	□	□	▨	■	■	■	■	■	■	▨	□	□	□	□
S8	□	□	□	□	▨	■	■	■	■	■	■	■	■	▨	□	□	□
S9	□	□	□	□	▨	■	■	■	■	■	■	■	■	■	▨	□	□
S10	□	□	□	▨	■	■	■	■	■	■	■	■	■	■	■	▨	□
S11	□	□	□	▨	■	■	■	■	■	■	■	■	■	▨	▨	▨	□
S12	□	□	▨	■	■	■	■	■	■	■	▨	▨	▨	□	□	□	□
S13	□	□	▨	■	■	■	■	▨	▨	▨	□	□	□	□	□	□	□
S14	□	▨	■	■	▨	▨	▨	□	□	□	□	□	□	□	□	□	□
S15	□	▨	▨	▨	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2 A

FIG. 19

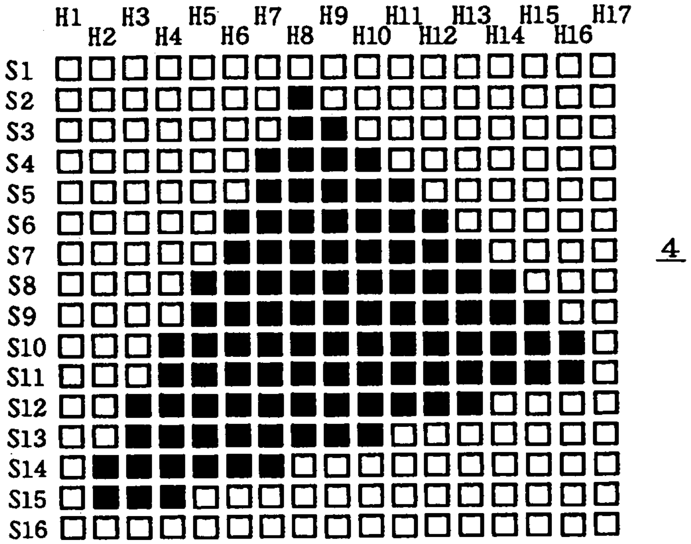

H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
H17
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
4

FIG. 20

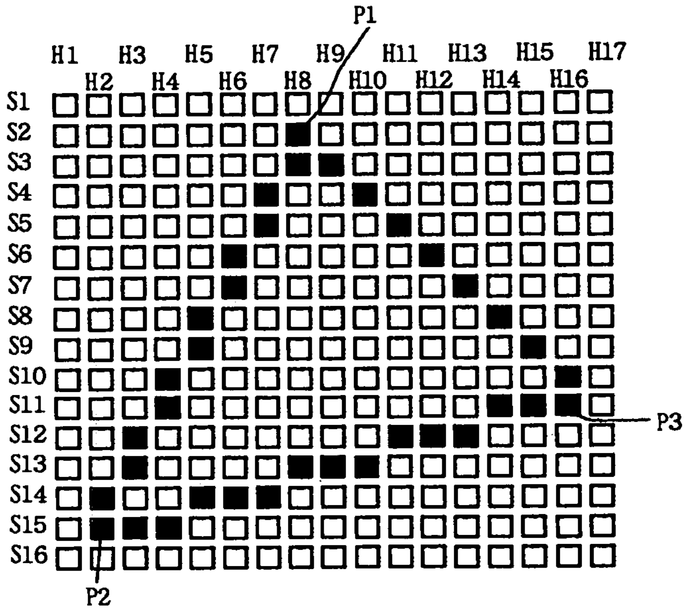

P1
H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
H17
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
P3
P2

FIG. 21

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S15	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2 A

FIG. 22

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2 A

FIG. 23

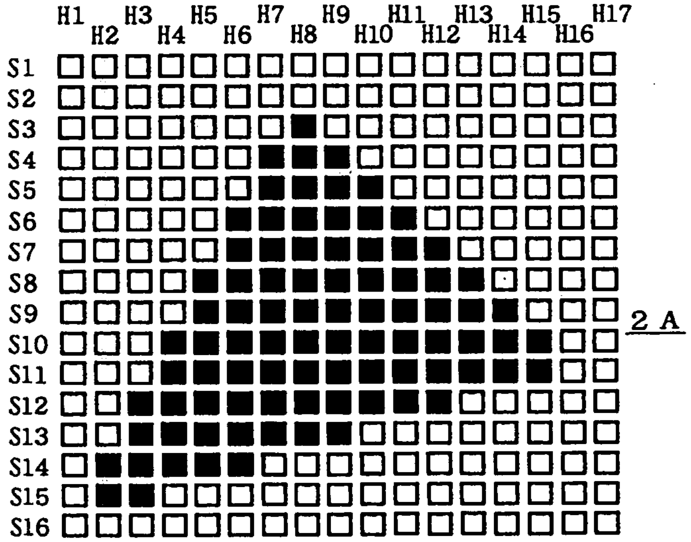

H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
H17
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
2 A

FIG. 24

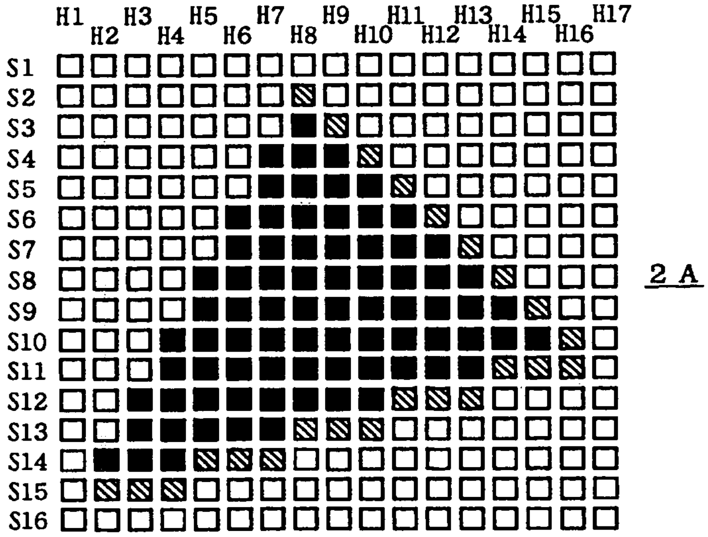

H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
H17
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
2 A

FIG. 25

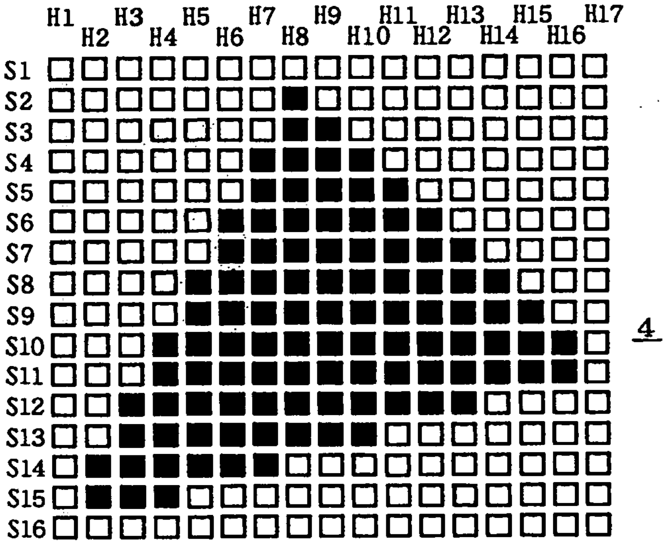

H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
H17
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
4

FIG. 26

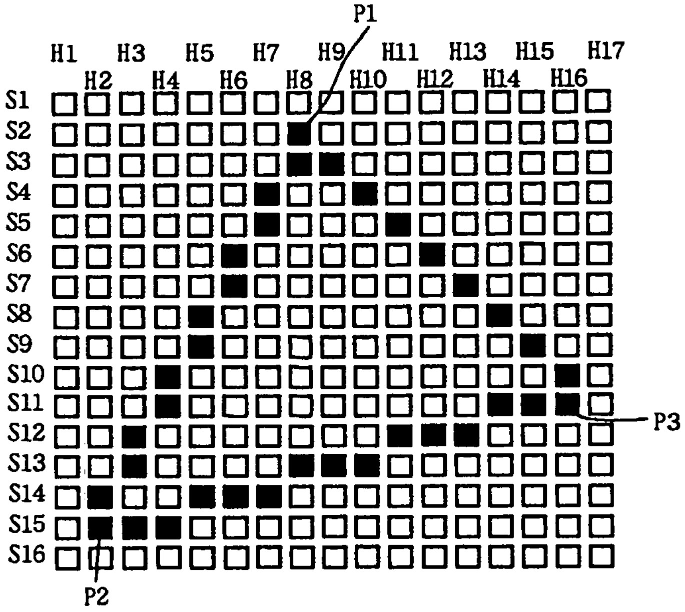
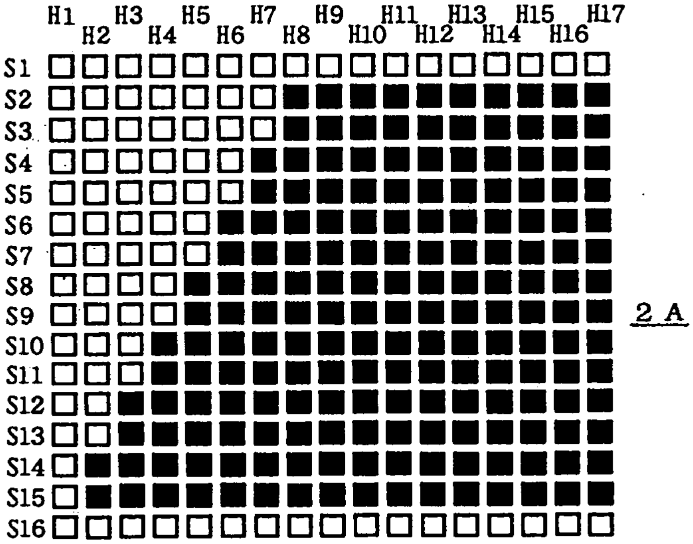
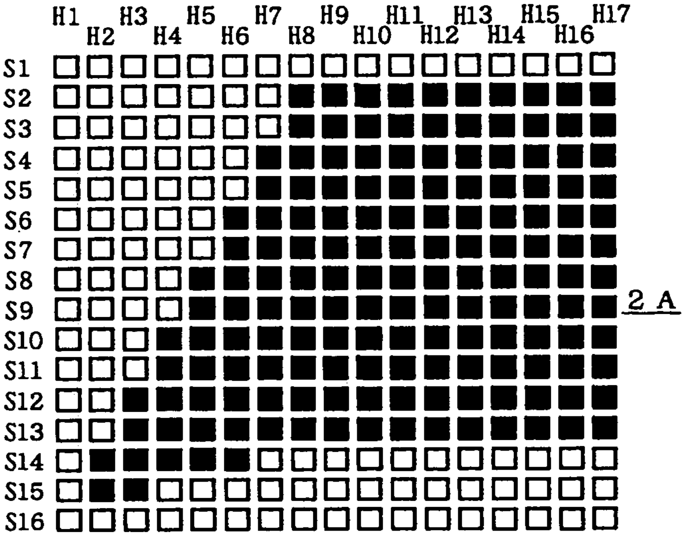

P1
H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
H17
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
P3
P2

FIG. 27



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S15	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 28



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 29

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	□	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	□	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	□	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2 A

FIG. 30

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	□	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	□	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	□	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

4

FIG. 31

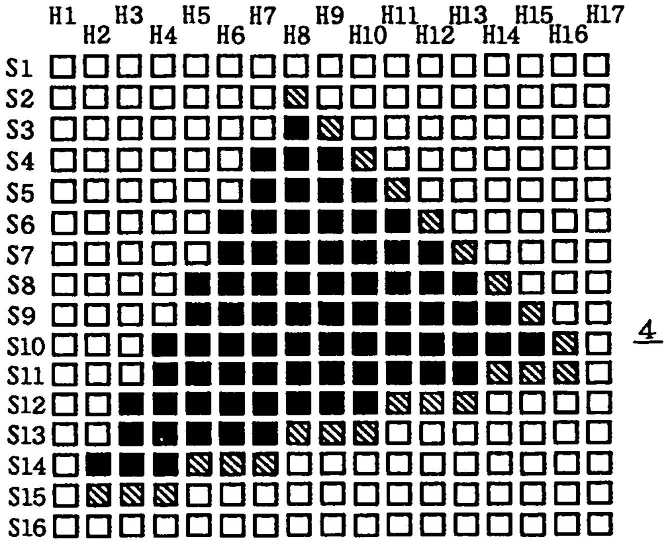
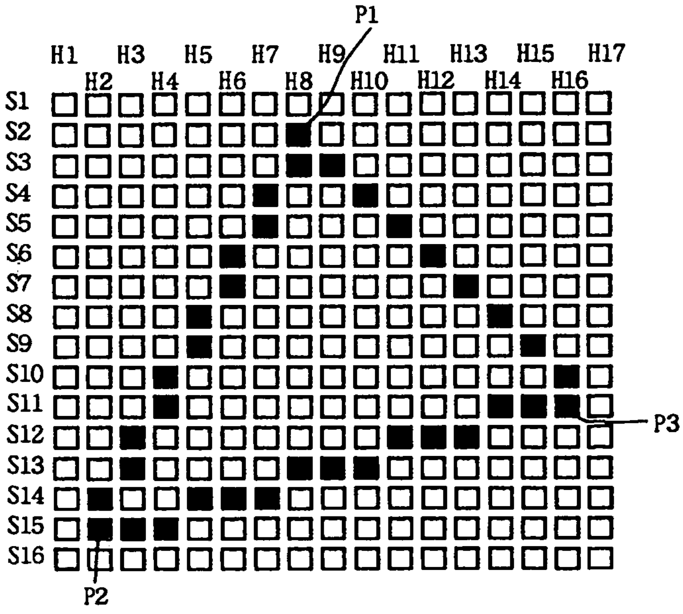
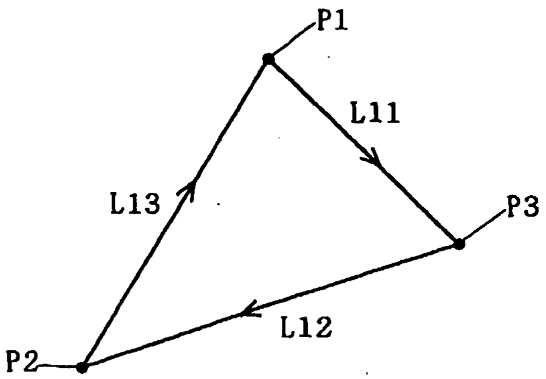

H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11 H12 H13 H14 H15 H16 H17
S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15 S16
4

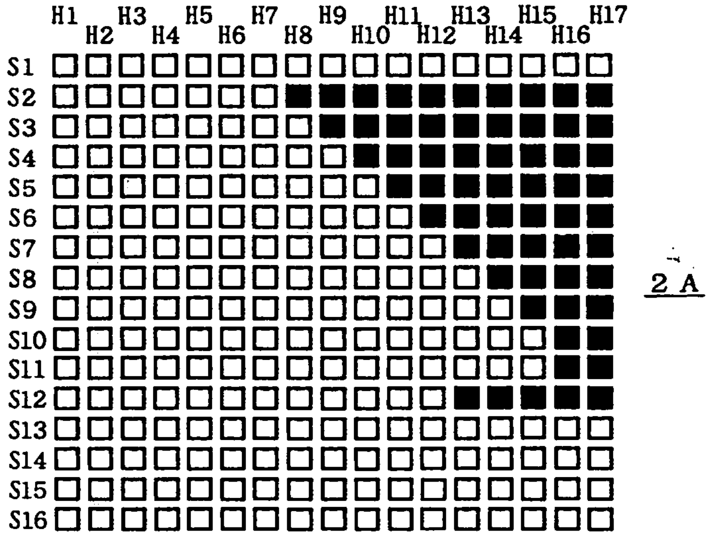
FIG. 32


P1
H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11 H12 H13 H14 H15 H16 H17
S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15 S16
P3
P2

F I G. 3 3

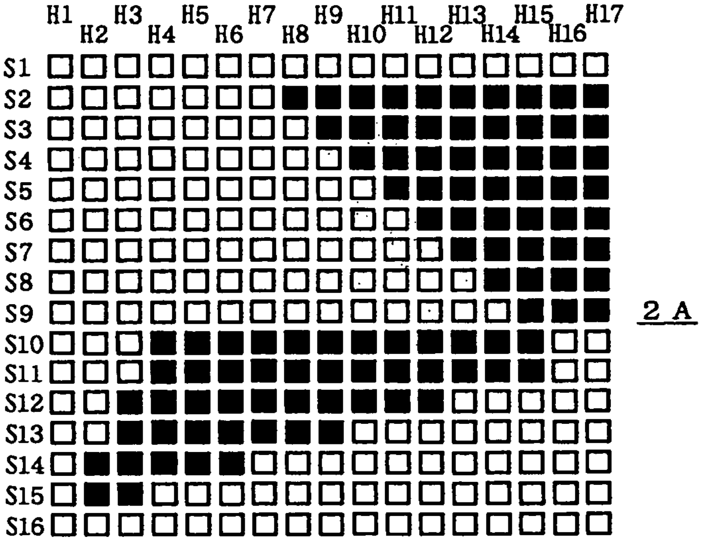


F I G. 3 4



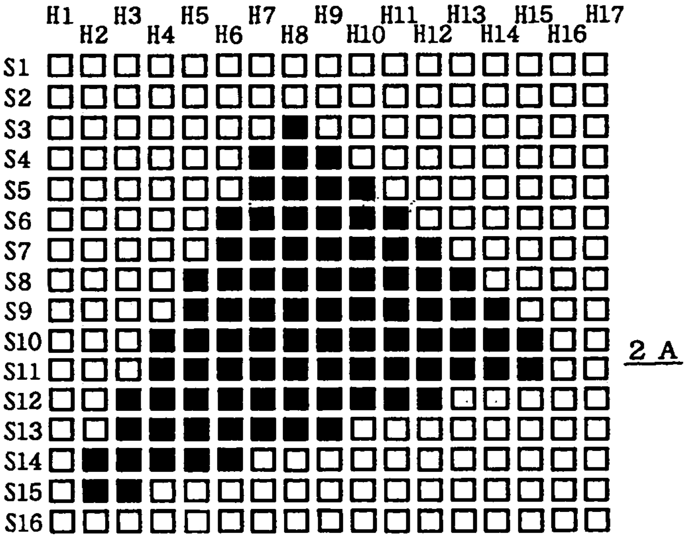
	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■
S6	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S7	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S10	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■
S11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■
S12	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■
S13	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S14	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S15	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

F I G. 3 5



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■
S6	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S7	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

F I G. 3 6



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	□	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	□	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	□	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 37

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	▨	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	▨	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	▨	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	▨	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	▨	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	▨	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	▨	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	▨	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	▨	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	▨	▨	▨	□
S12	□	□	■	■	■	■	■	■	■	■	▨	▨	▨	□	□	□	□
S13	□	□	■	■	■	■	■	▨	▨	▨	□	□	□	□	□	□	□
S14	□	■	■	■	▨	▨	▨	□	□	□	□	□	□	□	□	□	□
S15	□	▨	▨	▨	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

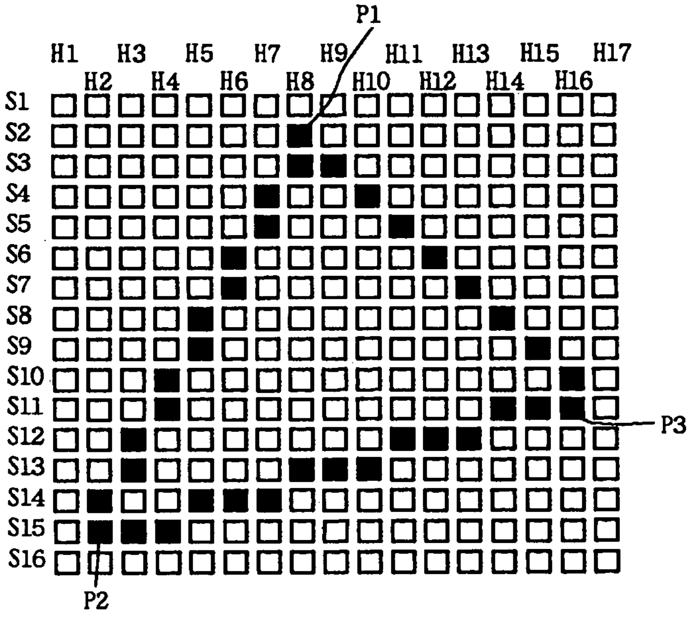
2 A

FIG. 38

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	■	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	■	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

4

F I G. 3 9


P1
H1 H3 H5 H7 H9 H11 H13 H15 H17
H2 H4 H6 H8 H10 H12 H14 H16
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
P2
P3

F I G. 4 0

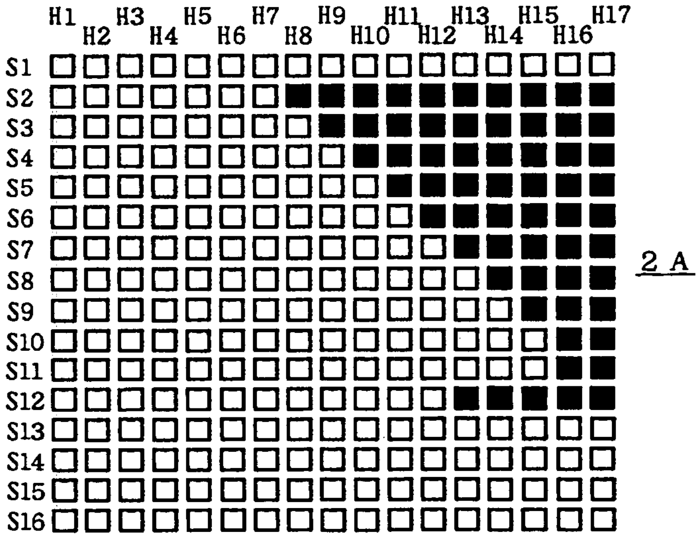

H1 H3 H5 H7 H9 H11 H13 H15 H17
H2 H4 H6 H8 H10 H12 H14 H16
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
2 A

FIG. 41

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■
S6	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S7	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2 A

FIG. 42

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	□	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	□	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	□	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2 A

FIG. 43

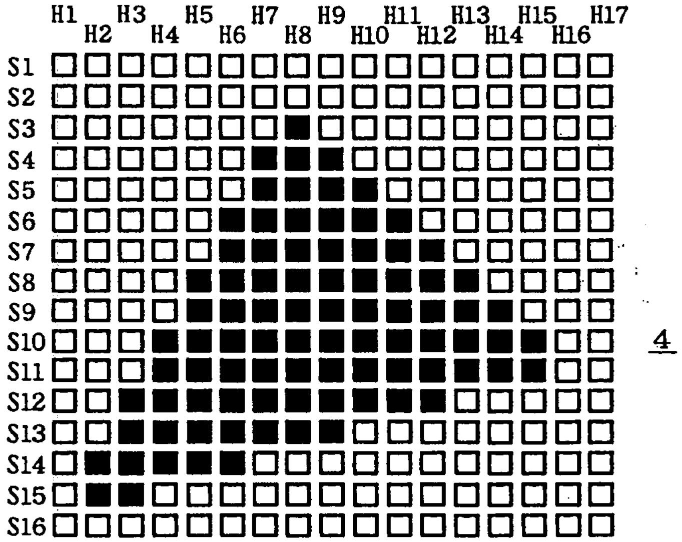

H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
H17
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
4

FIG. 44

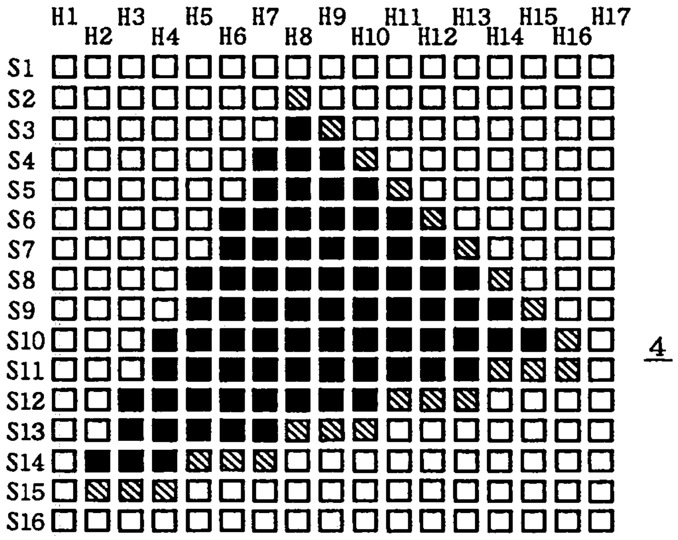

H1
H2
H3
H4
H5
H6
H7
H8
H9
H10
H11
H12
H13
H14
H15
H16
H17
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
4

FIG. 45

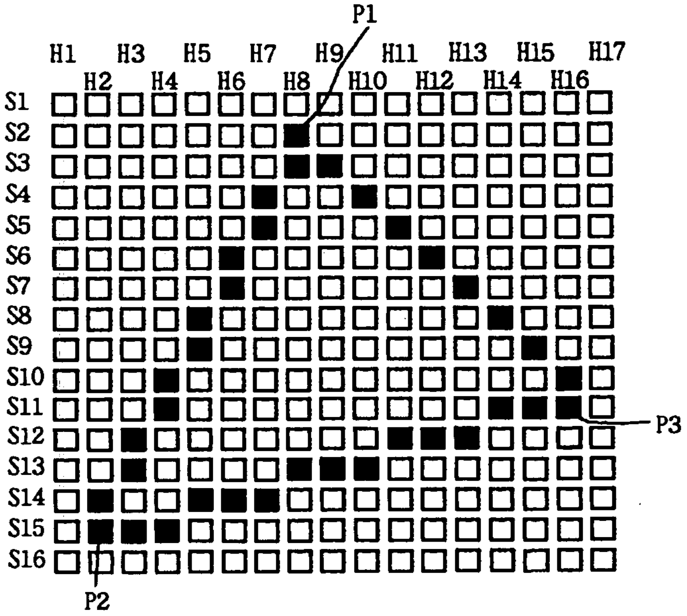

P1
H1 H3 H5 H7 H9 H11 H13 H15 H17
H2 H4 H6 H8 H10 H12 H14 H16
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
P3
P2

FIG. 46

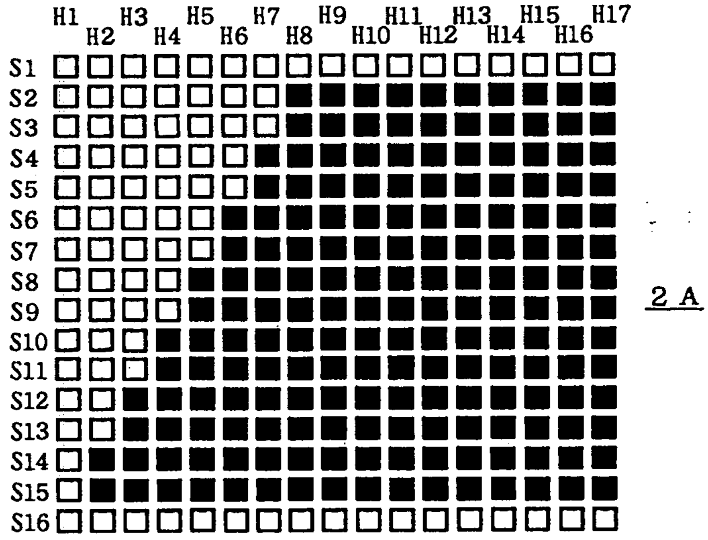
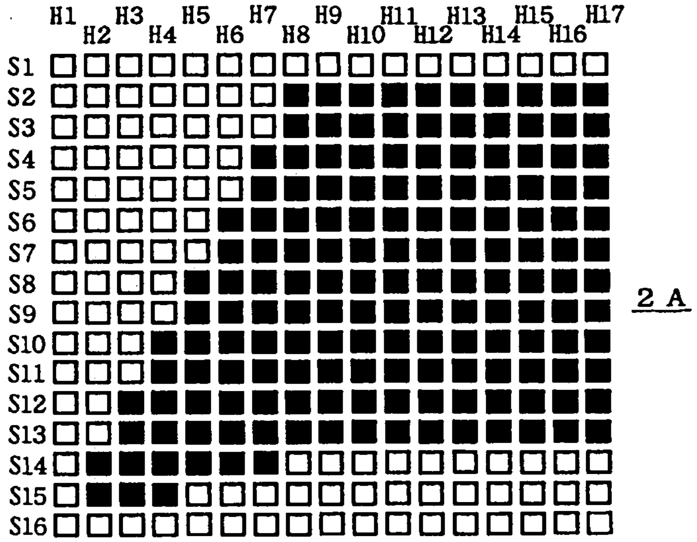
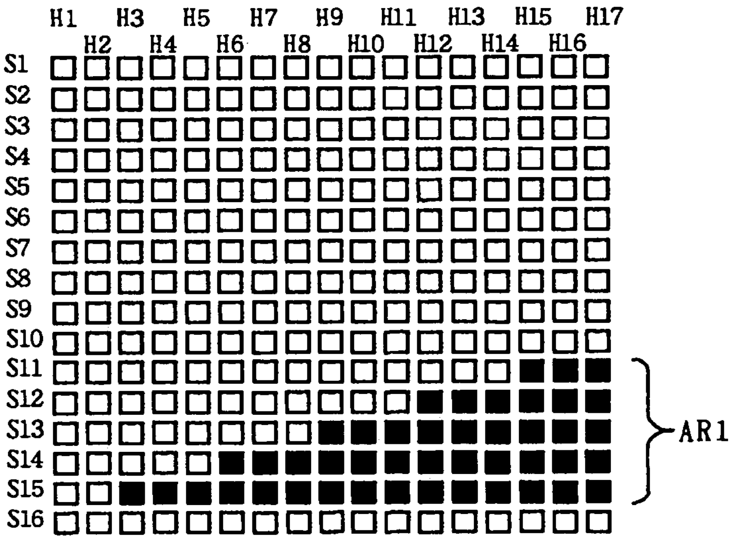

H1 H3 H5 H7 H9 H11 H13 H15 H17
H2 H4 H6 H8 H10 H12 H14 H16
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
2 A

FIG. 47



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 48



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S6	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S7	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S10	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S12	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S13	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S14	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S15	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 49

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S6	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■
S7	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■
S10	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■
S11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■
S12	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S13	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S14	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S15	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

AR2

FIG. 50

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	■	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	■	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2 A

FIG. 51

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	■	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	■	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

4

FIG. 52

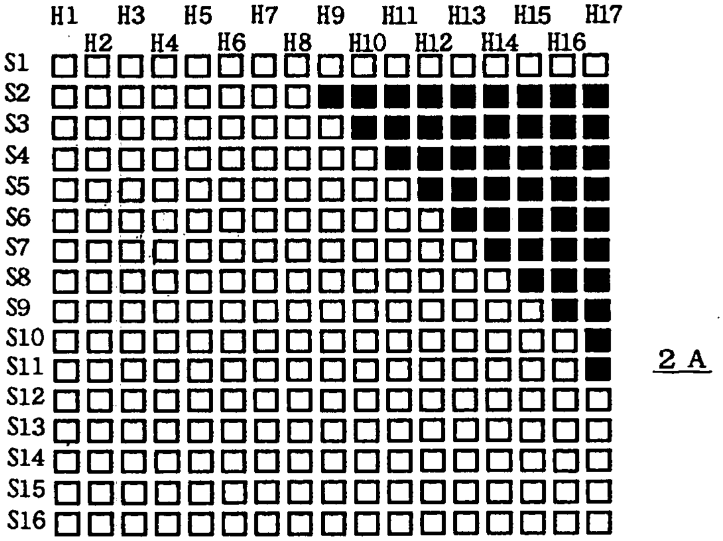
	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	□	□	■	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	□	□	□	■	□	□	□	□	□	□
S6	□	□	□	□	□	■	□	□	□	□	□	■	□	□	□	□	□
S7	□	□	□	□	□	■	□	□	□	□	□	□	■	□	□	□	□
S8	□	□	□	□	■	□	□	□	□	□	□	□	□	■	□	□	□
S9	□	□	□	□	■	□	□	□	□	□	□	□	□	□	■	□	□
S10	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	■	□
S11	□	□	□	■	□	□	□	□	□	□	□	□	□	■	■	■	□
S12	□	□	■	□	□	□	□	□	□	□	■	■	■	□	□	□	□
S13	□	□	■	□	□	□	□	■	■	■	□	□	□	□	□	□	□
S14	□	■	□	□	■	■	■	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

P1

P2

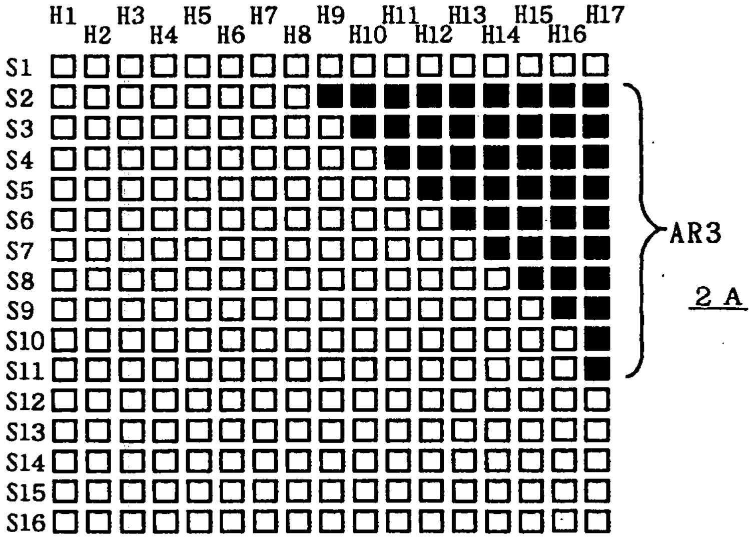
P3

F I G. 5 3



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S6	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■
S7	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■
S10	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■
S11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■
S12	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S13	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S14	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S15	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

F I G. 5 4



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S6	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■
S7	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■
S10	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■
S11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■
S12	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S13	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S14	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S15	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 55

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S6	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S7	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S10	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S12	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S13	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S14	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S15	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

AR4

FIG. 56

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■
S6	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■
S7	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■
S8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■
S9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	■	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2A

FIG. 57

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	
S2	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	
S3	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□	
S4	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□	
S5	□	□	□	□	□	□	■	■	■	■	■	□	□	□	□	□	□	
S6	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□	
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	□	□	□	□	
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□	2 A
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□	
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□	
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□	□	□	
S13	□	□	■	■	■	■	■	■	■	■	□	□	□	□	□	□	□	
S14	□	■	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	

FIG. 58

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	
S2	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	
S3	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□	
S4	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□	
S5	□	□	□	□	□	□	■	■	■	■	■	□	□	□	□	□	□	
S6	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□	
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	□	□	□	□	
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□	
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□	4
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	□	
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	□	□	□	□	
S13	□	□	■	■	■	■	■	■	■	■	□	□	□	□	□	□	□	
S14	□	■	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	

FIG. 59

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16
S1	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	■	□	■	□	□	□	□	□	■	□	□	□	□	□
S3	□	■	□	□	□	■	□	□	□	■	□	■	□	□	□	□
S4	■	□	□	□	□	□	■	□	■	□	□	□	■	□	□	□
S5	□	■	□	□	□	□	□	■	□	□	□	□	□	■	□	□
S6	□	□	■	□	□	□	■	□	□	□	□	□	□	□	■	□
S7	□	□	□	■	□	■	□	■	□	□	□	□	□	■	□	□
S8	□	□	□	□	■	□	□	□	■	□	□	□	■	□	□	□
S9	□	□	□	□	□	□	□	□	□	■	□	■	□	□	□	□
S10	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□

FIG. 60

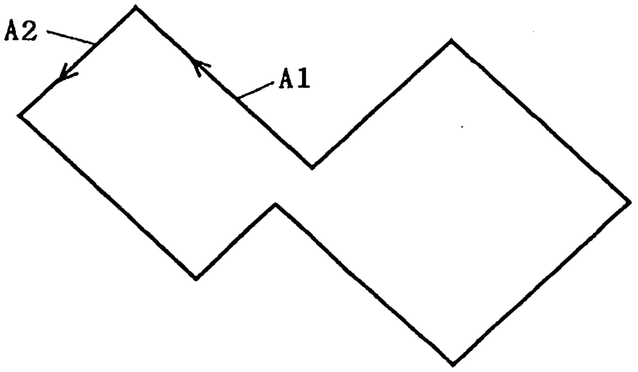
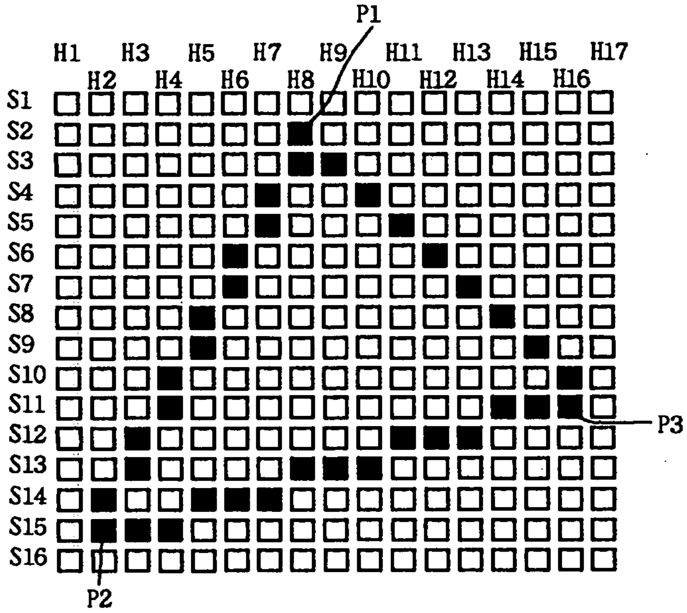


FIG. 61



	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	□	□	■	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	□	□	□	■	□	□	□	□	□	□
S6	□	□	□	□	□	■	□	□	□	□	□	■	□	□	□	□	□
S7	□	□	□	□	□	■	□	□	□	□	□	□	■	□	□	□	□
S8	□	□	□	□	■	□	□	□	□	□	□	□	□	■	□	□	□
S9	□	□	□	□	■	□	□	□	□	□	□	□	□	□	■	□	□
S10	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	■	□
S11	□	□	□	■	□	□	□	□	□	□	□	□	□	■	■	■	□
S12	□	□	■	□	□	□	□	□	□	□	■	■	■	□	□	□	□
S13	□	□	■	□	□	□	□	■	■	■	□	□	□	□	□	□	□
S14	□	■	□	□	■	■	■	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 62

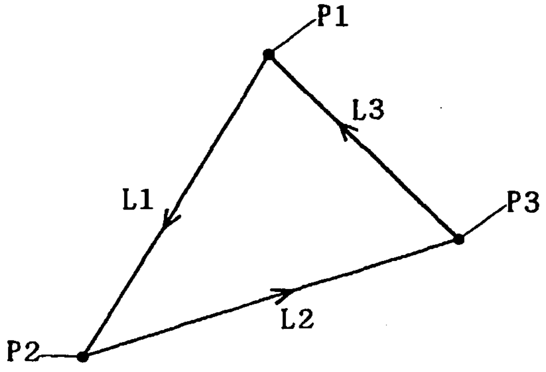


FIG. 63

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S12	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S13	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S14	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S15	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 64

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S15	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 65

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S3	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■
S4	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S5	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■
S6	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S7	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S12	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S13	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

FIG. 66

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17
S1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S3	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
S4	□	□	□	□	□	□	■	■	■	□	□	□	□	□	□	□	□
S5	□	□	□	□	□	□	■	■	■	■	□	□	□	□	□	□	□
S6	□	□	□	□	□	■	■	■	■	■	■	□	□	□	□	□	□
S7	□	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□
S8	□	□	□	□	■	■	■	■	■	■	■	■	■	□	□	□	□
S9	□	□	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□
S10	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S11	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	□	□
S12	□	□	■	■	■	■	■	■	■	■	■	■	□	□	□	□	□
S13	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□
S14	□	■	■	■	■	■	□	□	□	□	□	□	□	□	□	□	□
S15	□	■	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
S16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□